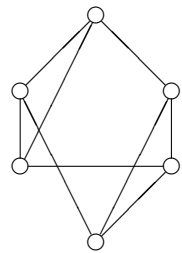
Hamiltonian Cycles and Shortest Path Problems: Learning Graph Theory through Islands and Bridges

Hiroyuki Yamane
Academic assembly (Science), University of Toyama

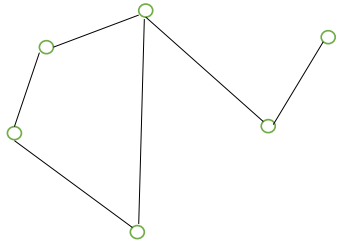
16/September/2026 (Wednesday), pm 14:00∼16:00,
Faculty of Science, Room B121
Tomidai Global SciFrontiers, Summer School 2026
2nd Summer Program, Interdisciplinary Thinking: Silos to Synergy,
Sept 14-18, 2026, University of Toyama

：
(1) グラフの用語の説明
(2) ハミルトン閉路とグレイコード
(3) ディラックの定理の紹介
(4) オーレの定理の紹介
(5) あみだくじのハミルトン閉路
(6) 最短重み経路

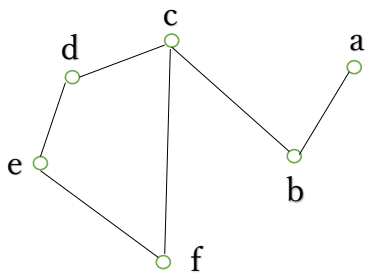
By a graph in this course, we do not mean a "bar graph" or a "pie chart." Instead, we consider graphs formed by connecting vertices $\circ$ with edges ——.



A point $\circ$ is called a vertex, and a line —— is called an edge.

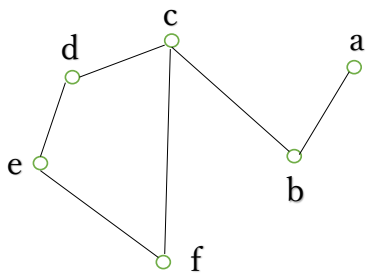


We sometimes assign labels to the vertices.



Let V denote the set of vertices of this graph. Then $\mathrm{V} = \{a, b, c, d, e, f\}$.

Vertices are sometimes assigned labels.



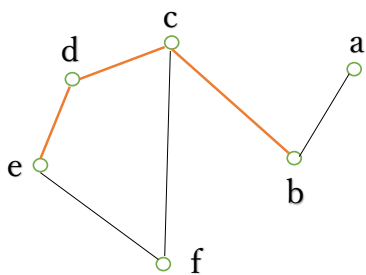
Let V be the set of vertices of this graph. Then $\mathrm{V} = \{a, b, c, d, e, f\}$.
An edge joining two vertices x and y is denoted by $\{x, y\}$. We have $\{x, y\} = \{y, x\}$.
Let E be the set of edges of this graph. Then

$$\mathrm{E} = \{\{a, b\}, \{b, c\}, \{c, d\}, \{c, f\}, \{d, e\}, \{e, f\}\}.$$

A graph G is an ordered pair (V, E), where V is the set of vertices and E is the set of edges.

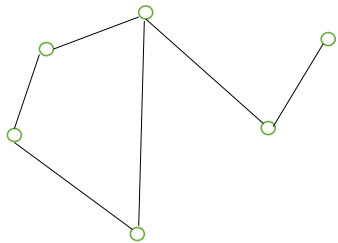
Path: A path joining two vertices x and y and passing through the vertices $z_1, z_2, \ldots, z_n$ is denoted by

$$\mathrm{x}z_1 z_2 \cdots z_n y.$$



The path highlighted in red is denoted by bcde.

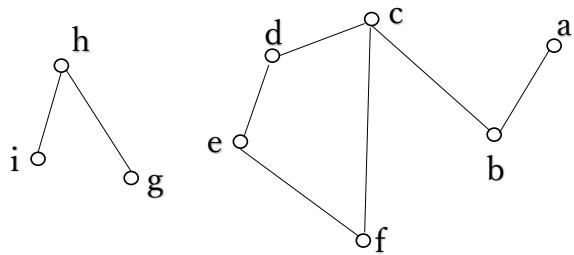
Let $G = (V, E)$ be a graph. If every pair of distinct vertices in G is connected by a path, then G is called a connected graph.



is a connected graph.

The graph shown below is not connected. It is said to be a disconnected graph.

$G =$



$G = (V, E)$,
$V = \{a, b, c, d, e, f, g, h, i\}$,
$E = \{\{a, b\}, \{b, c\}, \{c, d\}, \{c, f\}, \{d, e\}, \{e, f\}, \{g, h\}, \{h, i\}\}$.

Let $G = (V, E)$ be a graph. A connected subgraph of G is called a connected component of G.

A rigorous definition is as follows.
Let $x \in V$.

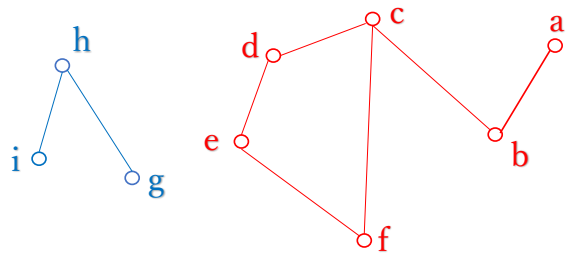
$$V(x) := \{\, y \in V \mid x \text{ and } y \text{ are connected by a path} \,\} \cup \{x\}.$$

$$E(x) := \{\{u, v\} \in E \mid u, v \in V(x)\}.$$

$$G(x) := (V(x), E(x)).$$

The subgraph $G(x)$ of G is called the connected component of G containing x.
The number of connected components of G is denoted by $k(G)$.

$G =$



There are two connected components; thus

$$k(G) = 2.$$

$$G(a) = G(b) = G(c) = G(d) = G(e) = G(f).$$

$$G(g) = G(h) = G(i).$$

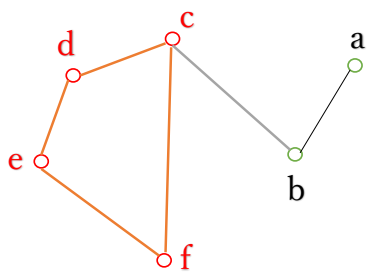
Cycle: A path that starts at a vertex z_1 and returns to z_1, passing through the vertices

$$z_1, z_2, \ldots, z_n, z_1,$$

is denoted by

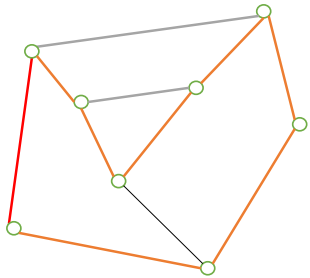
$$z_1 z_2 \cdots z_n z_1$$

and is called a cycle.



The cycle highlighted in red is denoted by $cdefc$.

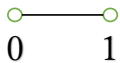
Hamiltonian Cycle: A cycle that visits every vertex exactly once is called a Hamiltonian cycle. A graph that contains a Hamiltonian cycle is called a Hamiltonian graph.



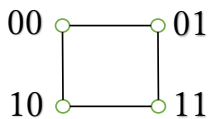
Applications of Hamiltonian Cycles — Gray Code:

The following first and second figures show the one and two bit Gray codes.

1-bit Gray code.

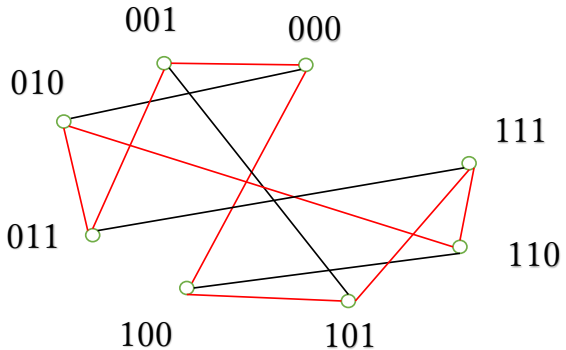


2-bit Gray code.



Applications of Hamiltonian Cycles — 3-Bit Gray Code:

3-bit Gray code.



The graph above is a Hamiltonian graph.

A Hamiltonian cycle is

$$000 \to 001 \to 011 \to 010 \to 110 \to 111 \to 101 \to 100 \to 000.$$

Notice that consecutive binary strings differ in exactly one bit.

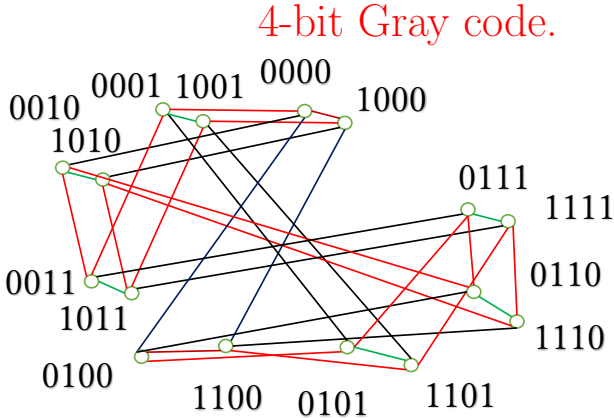
Such an ordering of binary strings is called a Gray code.

Gray codes are useful in digital systems because only one bit changes at a time, reducing errors during state transitions.

Gray codes were introduced by Frank Gray at Bell Laboratories in 1947. Gray's original motivation was to reduce errors in the conversion between analog signals and binary numbers. Because adjacent codewords differ in only one bit, Gray codes are widely used in communication systems, digital encoders, and error-sensitive switching circuits.

The 3-bit Gray code shown above is obtained from a Hamiltonian cycle of the 3-dimensional cube graph. More generally, a Hamiltonian cycle of the n-dimensional cube graph produces an n-bit Gray code.

Applications of Hamiltonian Cycles — 4-Bit Gray Code:



Theorem (Václav Chvátal (1973))

Let $G = (V, E)$ be a graph, where $V = \{v_1, \ldots, v_n\}$ is the vertex set and E is the edge set.

Let $1 \leq m < n$. Choose m vertices from $v_1, \ldots, v_n$ and relabel them as $w_1, \ldots, w_m$.

Define a subgraph $H = (W, F)$ of G by

$$W = \{w_1, \ldots, w_m\}, \qquad F = \{\{x, y\} \mid x, y \in W \text{ and } \{x, y\} \in E\}.$$

Then the following statement holds:

$$k(H) > n - m \Longrightarrow G \text{ is not Hamiltonian}$$

Historical Remark. The study of necessary conditions for Hamiltonian graphs has a long history. A particularly influential contribution was made by Václav Chvátal in 1973. For a graph G, let S be a set of vertices and let $k(G - S)$ denote the number of connected components of the graph obtained by removing the vertices of S. Chvátal introduced the notion of toughness and proved that every Hamiltonian graph satisfies

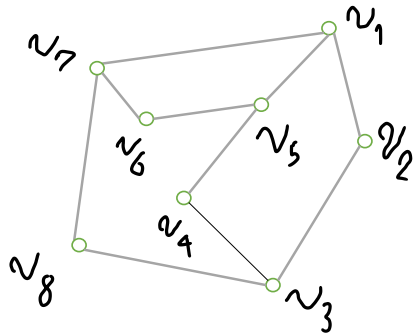
$$k(G - S) \leq |S|$$

for every vertex set S. The theorem above is a reformulation of this necessary condition. Indeed, if H is obtained by deleting $n - m$ vertices from G and satisfies

$$k(H) > n - m,$$

then G cannot contain a Hamiltonian cycle. Thus the theorem provides a simple and powerful method for proving that a graph is not Hamiltonian.

$G = (V, E) =$

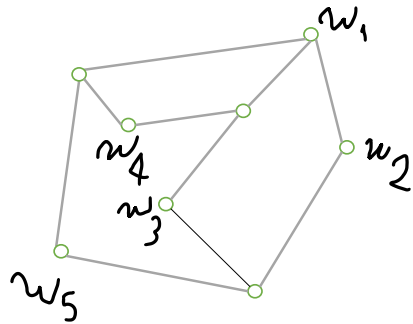


$V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$

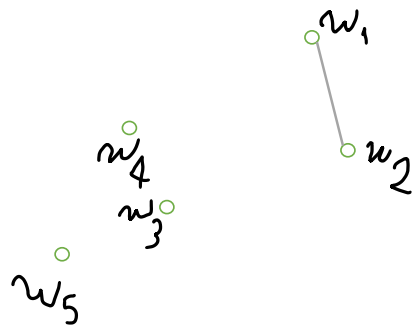
$E = \{\{v_1, v_2\}, \{v_1, v_5\}, \{v_1, v_7\}, \{v_2, v_3\}, \{v_3, v_4\}, \{v_3, v_8\},$
$\{v_4, v_5\}, \{v_5, v_6\}, \{v_6, v_7\}, \{v_7, v_8\}\}$

$n = 8$, $k(G) = 1$

Choose the vertices v_1, v_2, v_4, v_6, v_8 from V,
and denote them by w_1, w_2, w_3, w_4, w_5.



We obtain the following induced subgraph $H = (W, F)$.

$H = (W, F) =$ 

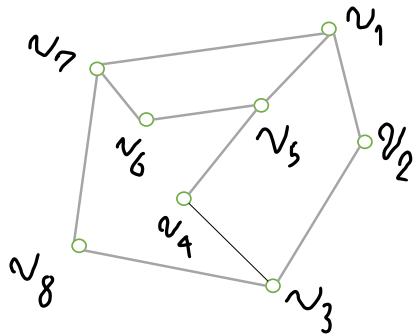
$$W = \{w_1, w_2, w_3, w_4, w_5\},\ F = \{\{w_1, w_2\}\},\ m = 5,\ k(H) = 4.$$

Since

$$n - m = 8 - 5 = 3 < 4 = k(H),$$

we have $k(H) > n - m$. Therefore, by the theorem, G is not a Hamiltonian graph.

Degree: Let $G = (V, E)$ be a graph. For a vertex $v \in V$, the number of vertices $w \in V$ that are adjacent to v (that is, $\{v, w\} \in E$) is called the degree of v and is denoted by $d(v)$.



$d(v_1) = 3$, $d(v_2) = 2$, $d(v_3) = 3$, $d(v_4) = 2$, $d(v_5) = 3$, $d(v_6) = 2$, $d(v_7) = 3$, $d(v_8) = 2$.

Dirac's Theorem (1952)

Let $G = (V, E)$ be a graph, where $V = \{v_1, \ldots, v_n\}$ is the vertex set and E is the edge set. Assume that $n \geq 3$.

Then the following statement holds.

$$\text{If} \quad d(v_i) \geq \frac{n}{2} \qquad (1 \leq i \leq n),$$

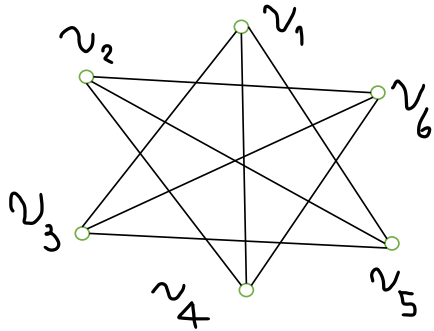
then G is a Hamiltonian graph.

Historical Remark. This theorem was proved by Gabriel Andrew Dirac in 1952. It is one of the most famous results in graph theory and provides a simple sufficient condition for the existence of a Hamiltonian cycle. The theorem states that if every vertex is connected to at least half of the vertices of the graph, then the graph is guaranteed to contain a Hamiltonian cycle. Dirac's theorem was one of the earliest general results giving a practical criterion for determining whether a graph is Hamiltonian. It has inspired much subsequent research on Hamiltonian cycles and graph connectivity.

Later, several important extensions and refinements were obtained, including results due to Øystein Ore, whose theorem (1960) states that a graph is Hamiltonian whenever

$$d(u) + d(v) \geq n$$

for every pair of non-adjacent vertices u and v.



$n = 6$, $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$
$d(v_1) = d(v_2) = d(v_3) = d(v_4) = d(v_5) = d(v_6) = 3$

Palmer's Algorithm (cf. Palmer, 1997)

Let $G = (V, E)$ be a graph, where $V = \{v_1, \ldots, v_n\}$ is the vertex set.

Assume that $G = (V, E)$ satisfies the hypothesis of Dirac's theorem.

Step 0: If

$$\{v_i, v_{i+1}\} \in E \qquad (1 \leq i \leq n-1)$$

and

$$\{v_n, v_1\} \in E,$$

then no further action is necessary. Otherwise, proceed to Step 1.

Step 1: If

$$\{v_i, v_{i+1}\} \notin E,$$

reorder

$$v_1, \ldots, v_n$$

as

$$v_i, v_{i+1}, \ldots, v_n, v_1, \ldots, v_{i-1},$$

and relabel the resulting sequence as

$$v_1, \ldots, v_n.$$

If $\{v_n, v_1\} \notin E$, reorder $v_1, \ldots, v_n$ as $v_n, v_1, v_2, \ldots, v_{n-1}$, and again relabel the resulting sequence as $v_1, \ldots, v_n$.

In either case, we obtain

$$\{v_1, v_2\} \notin E,$$

and proceed to Step 2.

Step 2: There exists an integer i satisfying

$$3 \leq i \leq n-1,$$

such that

$$\{v_1, v_i\} \in E \quad \text{and} \quad \{v_2, v_{i+1}\} \in E.$$

Reorder

$$v_1, \ldots, v_n$$

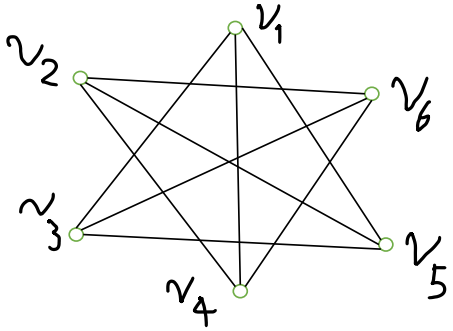
as

$$v_1,\ v_i,\ v_{i-1},\ \ldots,\ v_2,\ v_{i+1},\ v_{i+2},\ \ldots,\ v_n,$$

and relabel the resulting sequence as

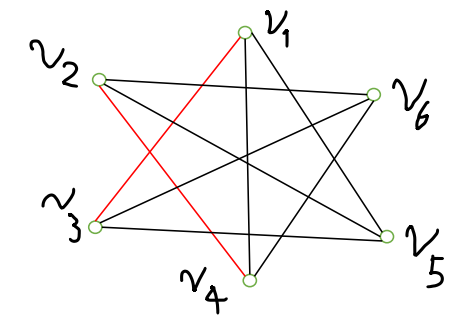
$$v_1, \ldots, v_n.$$

Then return to Step 0.

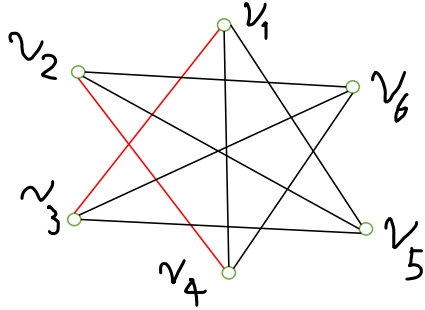


$v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5 \quad v_6$

Since $\{v_1, v_2\} \notin E$, the current ordering does not form a Hamiltonian cycle. We therefore apply Step 2 of Palmer's algorithm.

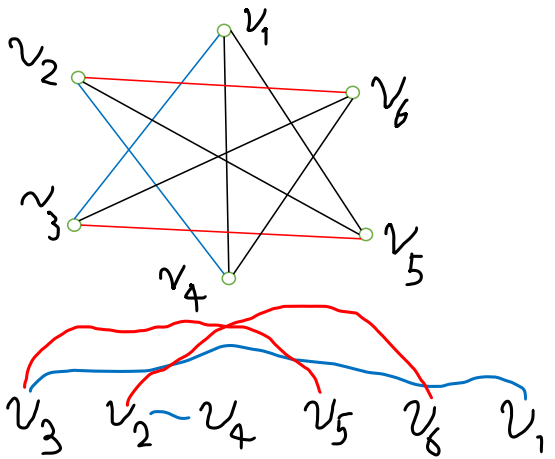


Reverse the order of v_2 and v_3.

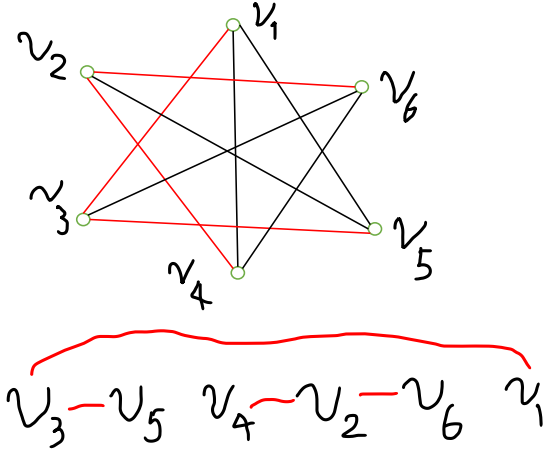


$v_1 - v_3 \quad v_2 - v_4 \quad v_5 \quad v_6$

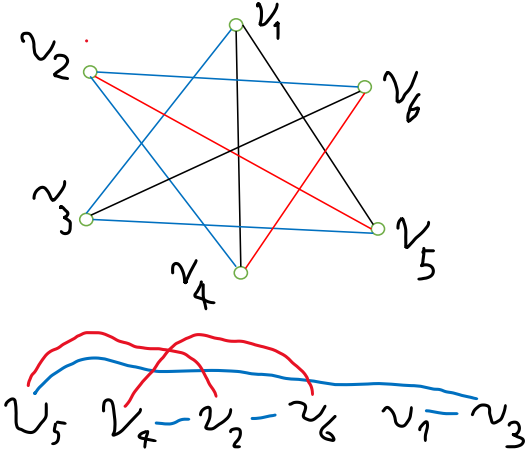
Apply Step 1: reorder the vertices so that v_3 becomes the first vertex and v_1 becomes the last vertex.



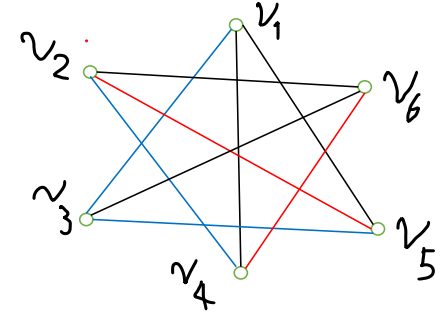
Apply Step 2: replace the sequence v_2, v_4, v_5 with v_5, v_4, v_2.

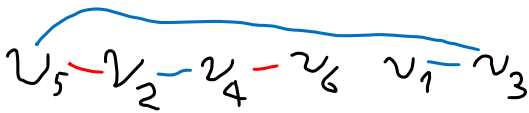


Apply Step 1: move v_5 to the left end and v_3 to the right end.

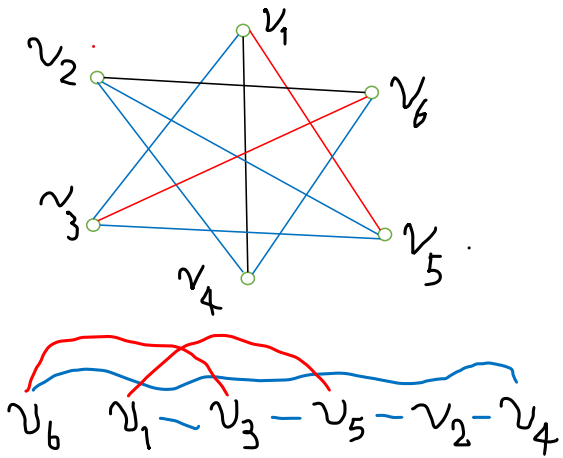


Apply Step 2: reverse the order of v_4 and v_2.

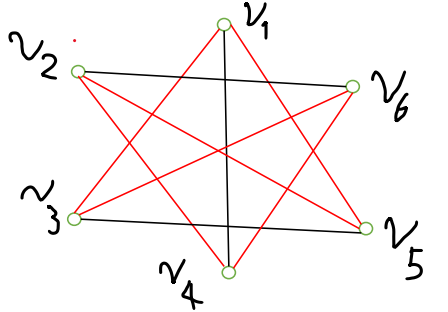




Apply Step 1: reorder the vertices so that v_6 becomes the first vertex and v_4 becomes the last vertex.



Apply Step 2: reverse the order of v_1 and v_3.



$v_6 - v_3 - v_1 - v_5 - v_2 - v_4$

Finish: we have gotten a Hamiltonian cycle.

Palmer's Algorithm (cf. Palmer, 1997)

Let $G = (V, E)$ be a graph, where $V = \{v_1, \ldots, v_n\}$ is the vertex set.

Assume that $G = (V, E)$ satisfies the hypothesis of Dirac's theorem.

Step 0: If

$$\{v_i, v_{i+1}\} \in E \qquad (1 \leq i \leq n-1)$$

and

$$\{v_n, v_1\} \in E,$$

then no further action is necessary. Otherwise, proceed to Step 1.

Step 1: If

$$\{v_i, v_{i+1}\} \notin E,$$

reorder

$$v_1, \ldots, v_n$$

as

$$v_i, v_{i+1}, \ldots, v_n, v_1, \ldots, v_{i-1},$$

and relabel the resulting sequence as

$$v_1, \ldots, v_n.$$

If $\{v_n, v_1\} \notin E$, reorder $v_1, \ldots, v_n$ as $v_n, v_1, v_2, \ldots, v_{n-1}$, and again relabel the resulting sequence as $v_1, \ldots, v_n$.

In either case, we obtain

$$\{v_1, v_2\} \notin E,$$

and proceed to Step 2.

Step 2: There exists an integer i satisfying

$$3 \leq i \leq n-1,$$

such that

$$\{v_1, v_i\} \in E \quad \text{and} \quad \{v_2, v_{i+1}\} \in E.$$

Reorder

$$v_1, \ldots, v_n$$

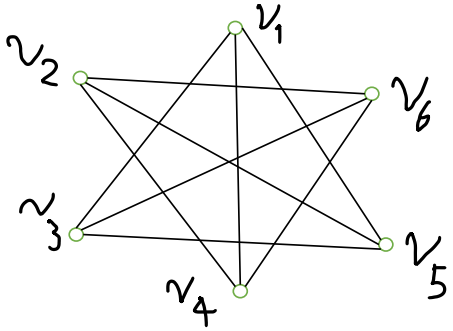
as

$$v_1,\ v_i,\ v_{i-1},\ \ldots,\ v_2,\ v_{i+1},\ v_{i+2},\ \ldots,\ v_n,$$

and relabel the resulting sequence as

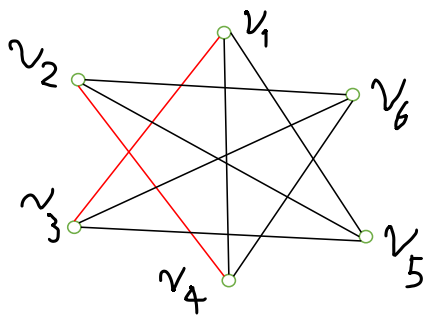
$$v_1, \ldots, v_n.$$

Then return to Step 0.

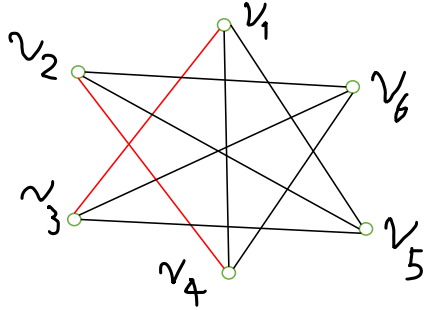


$v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5 \quad v_6$

Since $\{v_1, v_2\} \notin E$, the current ordering does not form a Hamiltonian cycle. We therefore apply Step 2 of Palmer's algorithm.

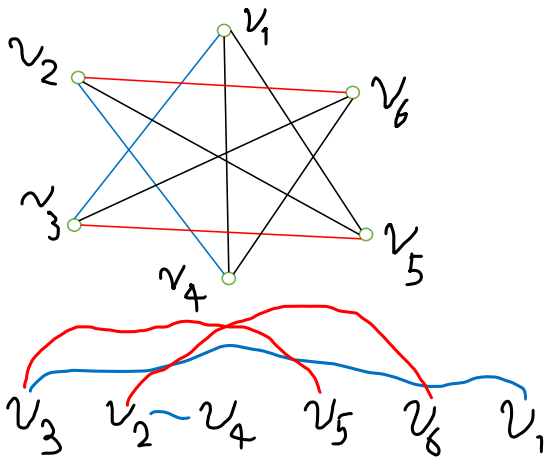


Reverse the order of v_2 and v_3.

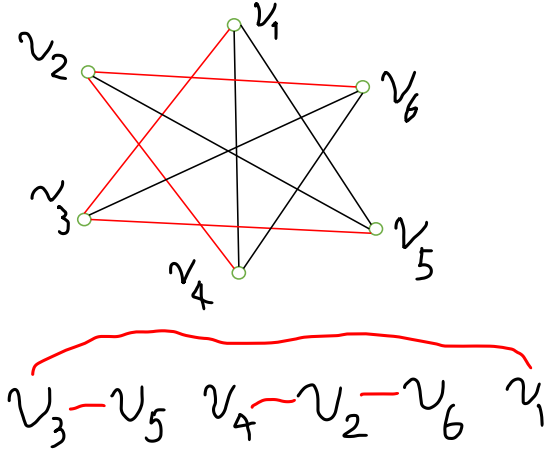


$v_1 - v_3 \quad v_2 - v_4 \quad v_5 \quad v_6$

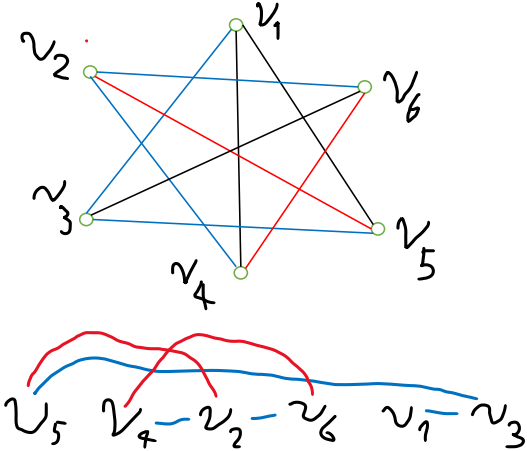
Apply Step 1: reorder the vertices so that v_3 becomes the first vertex and v_1 becomes the last vertex.



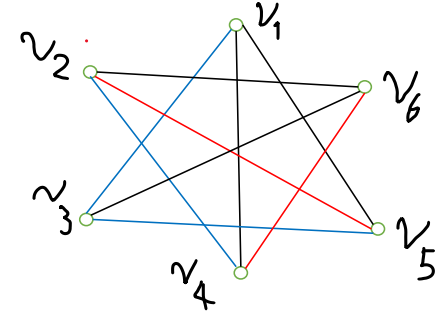
Apply Step 2: replace the sequence v_2, v_4, v_5 with v_5, v_4, v_2.



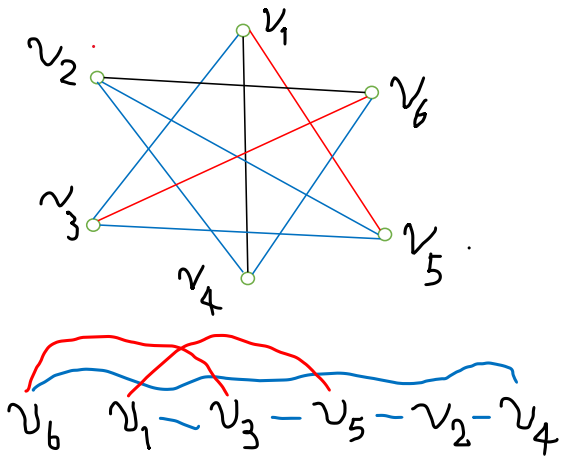
Apply Step 1: move v_5 to the left end and v_3 to the right end.



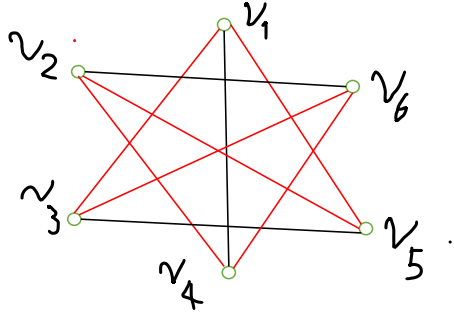
Apply Step 2: reverse the order of v_4 and v_2.



Apply Step 1: reorder the vertices so that v_6 becomes the first vertex and v_4 becomes the last vertex.



Apply Step 2: reverse the order of v_1 and v_3.



$v_6 - v_3 - v_1 - v_5 - v_2 - v_4$

Finish: we have gotten a Hamiltonian cycle.

Ore's Theorem (Ore, 1960)

Let $G = (V, E)$ be a graph, where $V = \{v_1, \ldots, v_n\}$ is the vertex set and E is the edge set. Assume that $n \geq 3$.

Suppose that for every pair of vertices $1 \leq i < j \leq n$,

$$\{v_i, v_j\} \notin E \qquad \Longrightarrow \qquad d(v_i) + d(v_j) \geq n.$$

Then

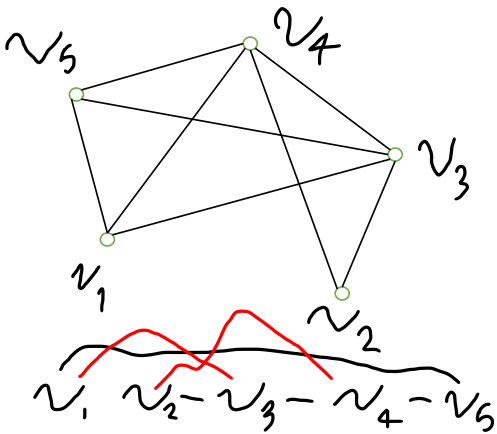
G is a Hamiltonian graph.

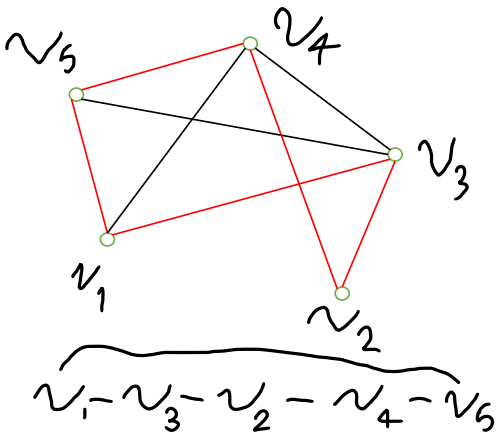
Historical Remark.

The theorem was proved by Øystein Ore in 1960.

It generalizes Dirac's theorem by replacing a lower bound on the degree of each vertex with a condition on the degree sum of non-adjacent vertices.

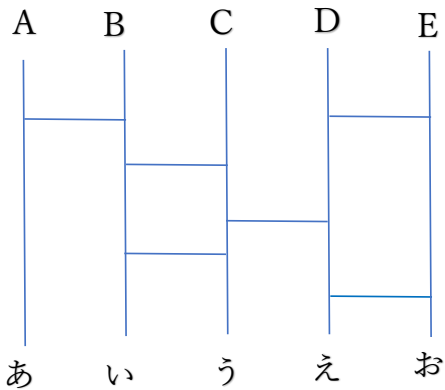
Ore's theorem is one of the classical sufficient conditions for the existence of a Hamiltonian cycle and has had a major influence on the development of Hamiltonian graph theory.


v_5
v_4
v_3
v_1
v_2
v_1 v_2 – v_3 – v_4 – v_5


v_5
v_4
v_3
v_1
v_2
$v_1 - v_3 - v_2 - v_4 - v_5$

Next, we describe an algorithm of J. H. Conway, N. J. A. Sloane, and A. R. Wilks (1989) (see also Rappaport–Strasser (1959) and Steinhaus–Johnson–Trotter algorithm (1963)) for constructing Hamiltonian cycles in the Cayley graphs of finite Coxeter groups.

Amidakuji (Ladder Lottery)



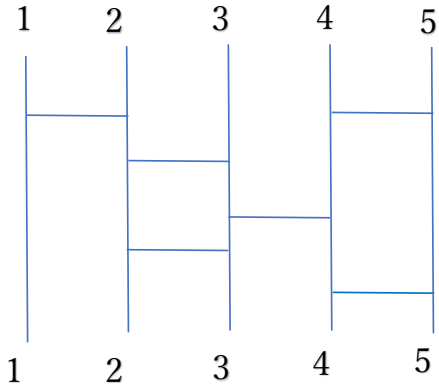
An *Amidakuji* (also called a *ladder lottery* or *ghost-leg*) determines a permutation of the numbers $1, 2, \ldots, n$.

Starting from the top, follow each vertical line downward. Whenever a horizontal bar is encountered, move across it and continue downward.

In this way, each starting position is matched with exactly one ending position, giving a permutation of $\{1, 2, \ldots, n\}$.

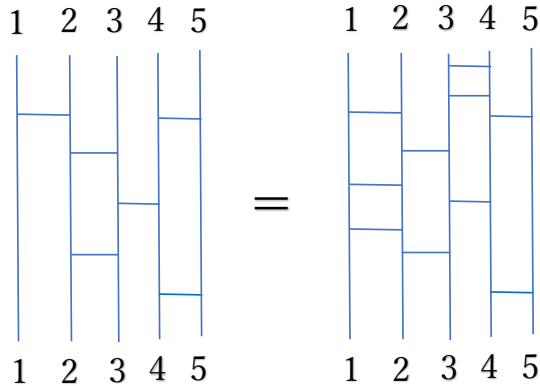
Thus every Amidakuji defines an element of the symmetric group S_n, which will be defined later.

We represent an Amidakuji by a two-line notation.



$$= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 1 & 3 & 4 & 2 \end{pmatrix}$$

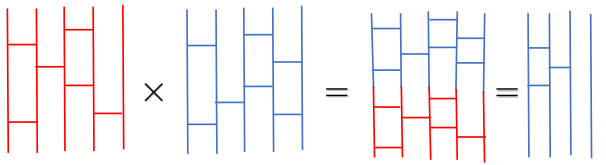
However, we regard two Amidakuji as the same whenever they determine the same destination for each starting position.



$$= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 1 & 3 & 4 & 2 \end{pmatrix}$$

An Amidakuji is determined entirely by the permutation that it represents. Different drawings may correspond to the same permutation.

We define the product of two Amidakuji by concatenating them.

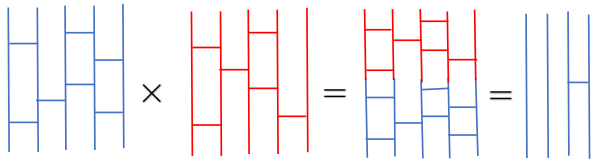


$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 2 & 3 & 1 & 4 \end{pmatrix} \times \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 2 & 4 & 5 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 2 & 1 & 4 & 5 \end{pmatrix}.$$

In terms of permutations, the product is obtained by first following the left Amidakuji and then the right Amidakuji.

Thus the set of all Amidakuji on n strands is closed under this operation.

The following show the 'opposite' product for the one of the above sheet.

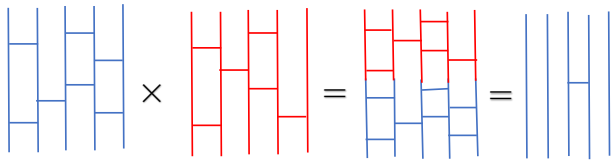


$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 2 & 4 & 5 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 2 & 3 & 1 & 4 \end{pmatrix}$$
$$= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 4 & 3 & 5 \end{pmatrix}$$

It means that generally the 'opposite' product is not the same as the original one.

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 2 & 3 & 1 & 4 \end{pmatrix} \times \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 2 & 4 & 5 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 2 & 1 & 4 & 5 \end{pmatrix}$$

Now let us multiply the same two Amidakuji in the opposite order.



$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 2 & 4 & 5 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 2 & 3 & 1 & 4 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 4 & 3 & 5 \end{pmatrix}.$$

This example shows that the order of multiplication matters.

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 2 & 3 & 1 & 4 \end{pmatrix} \times \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 2 & 4 & 5 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 2 & 1 & 4 & 5 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 2 & 4 & 5 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 2 & 3 & 1 & 4 \end{pmatrix}$$
$$= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 4 & 3 & 5 \end{pmatrix}.$$

Therefore, in general,

$$AB \neq BA.$$

That is, multiplication of Amidakuji (equivalently, permutations) is generally not commutative.

Let S_n denote the set of all Amidakuji with n vertical strands.

$$S_n = \left\{ \left(\begin{array}{cccc} 1 & 2 & \cdots & n \\ i_1 & i_2 & \cdots & i_n \end{array} \right) \;\middle|\; \begin{array}{l} 1 \leq i_1, i_2, \ldots, i_n \leq n, \\ i_1, i_2, \ldots, i_n \text{are distinct} \end{array} \right\}.$$

Since each element of S_n corresponds to a permutation of $1, 2, \ldots, n$, the number of elements of S_n is $n!$.

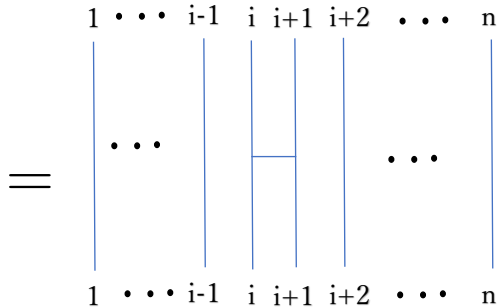
$$|S_n| = n!$$

The set S_n, together with the multiplication defined above, is called the symmetric group of degree n.

The term group is one of the fundamental concepts of modern algebra in mathematics.

The atoms (simple transpositions) of S_n are

$$s_i = \begin{pmatrix} 1 & \cdots & i-1 & i & i+1 & i+2 & \cdots & n \\ 1 & \cdots & i-1 & i+1 & i & i+2 & \cdots & n \end{pmatrix} \in S_n \qquad (1 \le i \le n-1).$$



The permutation s_i exchanges i and $i+1$ and fixes all other numbers. In terms of Amidakuji, s_i consists of a single horizontal bar joining the i-th and $(i+1)$-st strands.

The Cayley graph of the Amidakuji group S_n.

The vertex set is S_n, and the edge set is

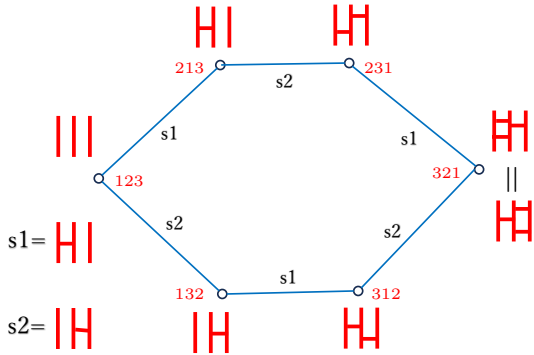
$$E = \{\{x, xs_i\} \mid x \in S_n,\ 1 \leq i \leq n-1\}.$$

The graph with vertex set S_n and edge set E is called the Cayley graph of S_n.

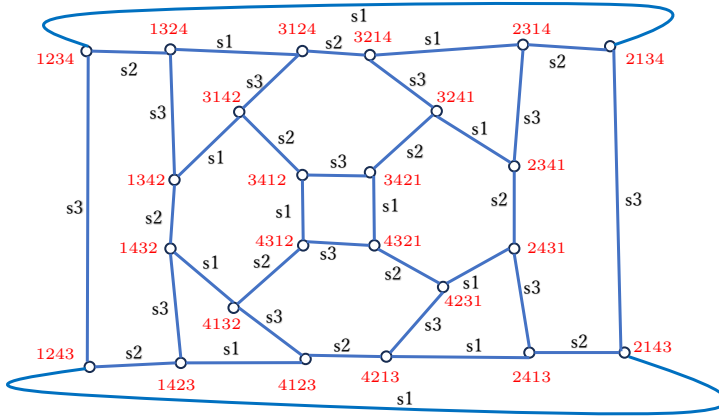
Two vertices x and y are joined by an edge if $y = xs_1$ or $y = xs_2$.

In other words, an edge corresponds to multiplying by a simple transposition.

Cayley graph of S_3

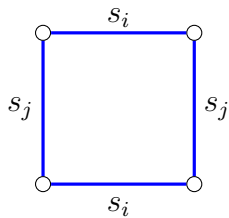


Cayley graph of S_4



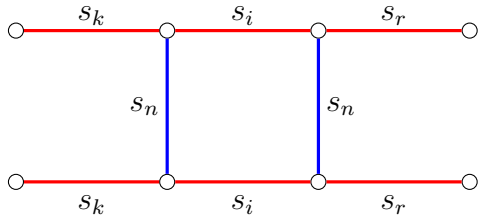
How to obtain a Hamiltonian cycle for S_n.

The crucial observation is that the generators s_i and s_j commute whenever $|i-j| \geq 2$, namely,

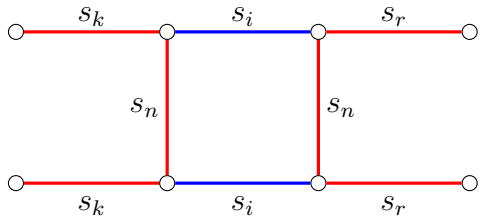


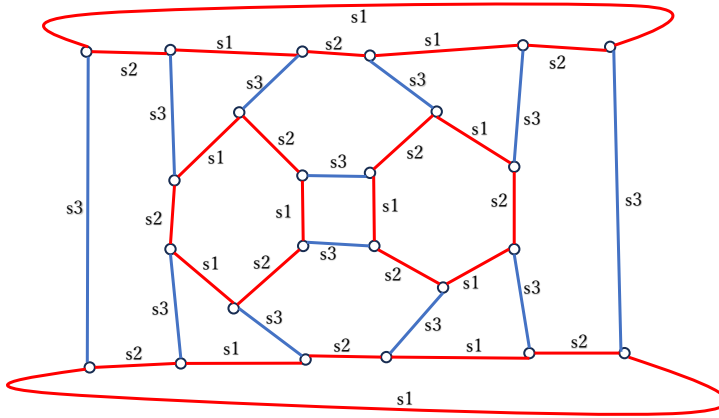
$(|i-j| \geq 2)$

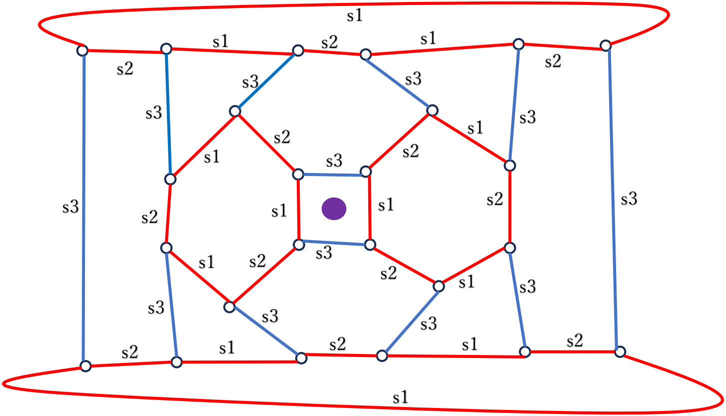
In particular, $|i-j| \geqq 2$, we have $s_i s_n = s_n s_i$ for $1 \leq i \leq n-2$. So we can have a Hamiltonian cycle on an induction on n in the way; find i with $1 \leq i \leq n-2$. Then

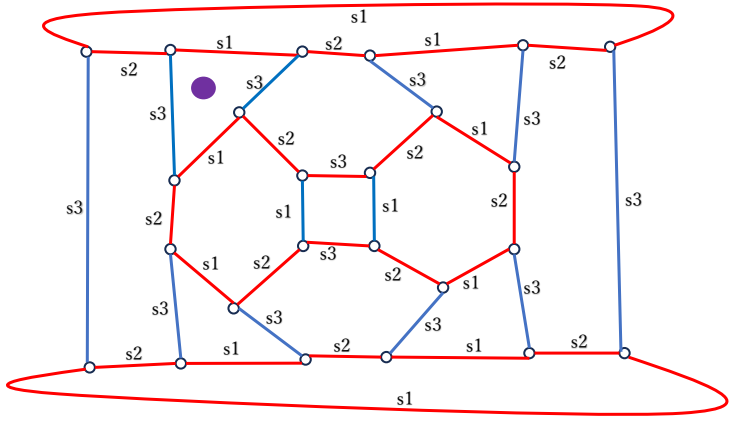


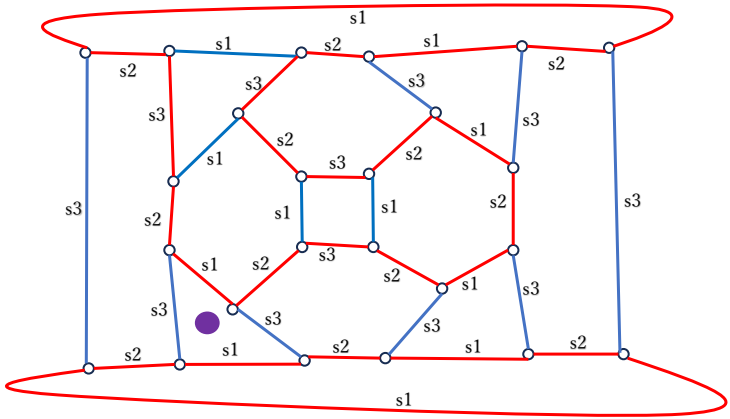
We cut the cycle along the two opposite edges labelled s_i and reconnect the resulting paths using the two edges labelled s_n. That is, the above configuration is replaced by the following one.

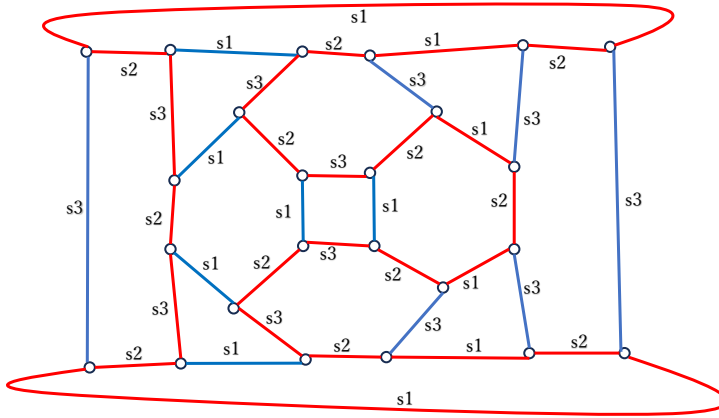



s1
s1
s2
s1
s2
s2
s3
s3
s3
s3
s1
s2
s3
s2
s1
s3
s2
s3
s1
s2
s1
s1
s2
s3
s1
s2
s2
s1
s3
s3
s3
s2
s1
s2
s2
s1
s1
s2
s3
s3
s1
s2
s3
s3
s2
s1
s2
s1
s2
s3
s3
s3
s2
s1
s2
s1
s2
s3
s3
s3
s2
s1


s1
s1
s2
s1
s2
s3
s3
s2
s3
s3
s1
s2
s3
s2
s1
s3
s3
s2
s1
s1
s2
s2
s3
s1
s2
s3
s2
s1
s3
s3
s3
s2
s1
s2
s1
s2
s1


s1
s1
s2
s1
s2
s2
s3
s3
s3
s3
s1
s2
s2
s1
s3
s3
s1
s2
s2
s1
s1
s3
s3
s2
s1
s2
s2
s1
s3
s3
s3
s3
s2
s1
s2
s1
s2
s1
s2
s3
s3
s1


s1
s1
s2
s1
s2
s2
s3
s3
s3
s3
s1
s2
s3
s2
s1
s3
s1
s2
s3
s2
s1
s1
s2
s3
s2
s1
s3
s3
s3
s2
s1
s2
s1
s2
s1
s2
s3
s3
s1



We got a Hamiltonian cycle!

Generalized Root Systems (Heckenberger-Yamane (2008), Cuntz-Heckenberger (2009), Yamane (2016))
Let $J_{x,y} := \{z \in \mathbb{Z} \,|\, x \le z \le y\}$. Fix $n \in \mathbb{N}$ and set

$$\mathfrak{A} = \{{}^t(a_1, \dots, a_n) \in \mathbb{R}^n \,|\, a_i \in \mathbb{Z} \ (\in J_{1,n})\}.$$

For $K = \{k_1, \dots, k_n\} \subset \mathfrak{A}$, we say that K is a $\mathbb{Z}$-basis if

$$\mathfrak{A} = \mathbb{Z}k_1 \oplus \cdots \oplus \mathbb{Z}k_n \quad (\text{i.e., } \det({}^tk_1, \dots, {}^tk_n) = \pm 1).$$

Let R be a subset of $\mathfrak{A}$. Let $\widetilde{\mathbb{B}}$ denote the set of all $\mathbb{Z}$-bases of $\mathfrak{A}$. For $\Pi \in \widetilde{\mathbb{B}}$, define

$$\mathfrak{A}_\Pi^\pm := \Big\{ \sum_{\alpha \in \Pi} n_\alpha \alpha \;\Big|\; n_\alpha \in \pm\mathbb{Z}_{\ge 0} \Big\}.$$

Define

$$\mathbb{B}_R := \Big\{ \Pi \in \widetilde{\mathbb{B}} \;\Big|\; R = (R \cap \mathfrak{A}_\Pi^+) \cup (R \cap \mathfrak{A}_\Pi^-) \Big\}.$$

Let $\mathbb{B}$ be a subset of $\mathbb{B}_R$. We say that $(R, \mathbb{B})$ is a **generalized root system** (on $\mathfrak{A}$) if $R \neq \emptyset$ and $\mathbb{B} \neq \emptyset$, and the following condition ($\sharp$) is satisfied.

"($\sharp$): For every $\Pi \in \mathbb{B}$ and every $\alpha \in \Pi$,

$$\mathbb{Z}\alpha \cap R = \{\alpha, -\alpha\},$$

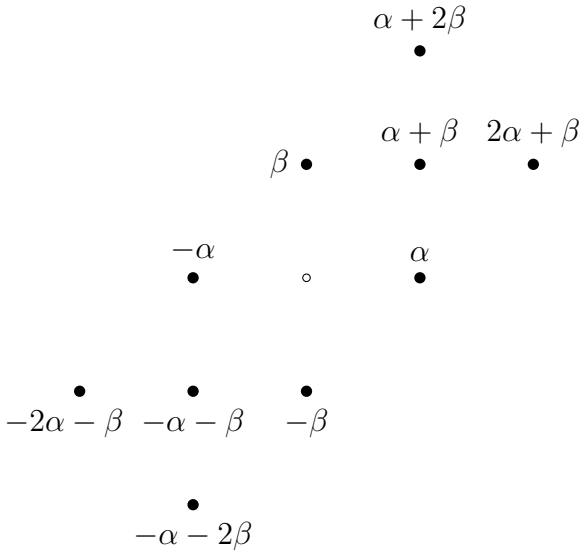
and there exists $\Pi^{(\alpha)} \in \mathbb{B}$ such that

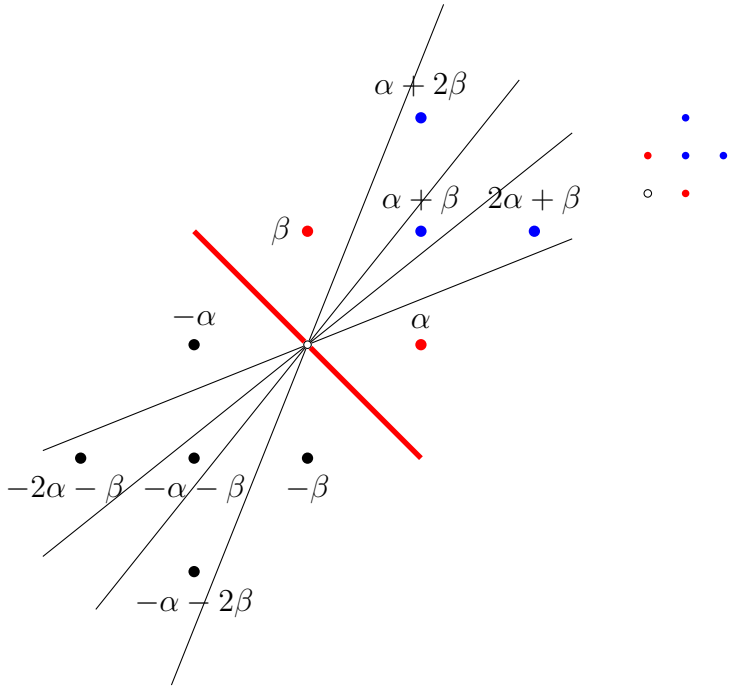
$$R \cap \mathfrak{A}^+_{\Pi} \cap \mathfrak{A}^-_{\Pi^{(\alpha)}} = \{\alpha\}.$$

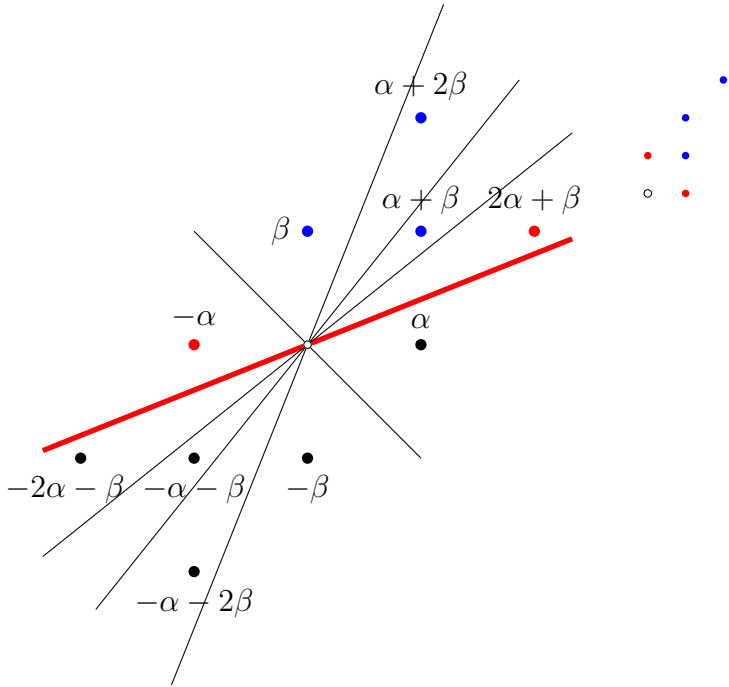
"

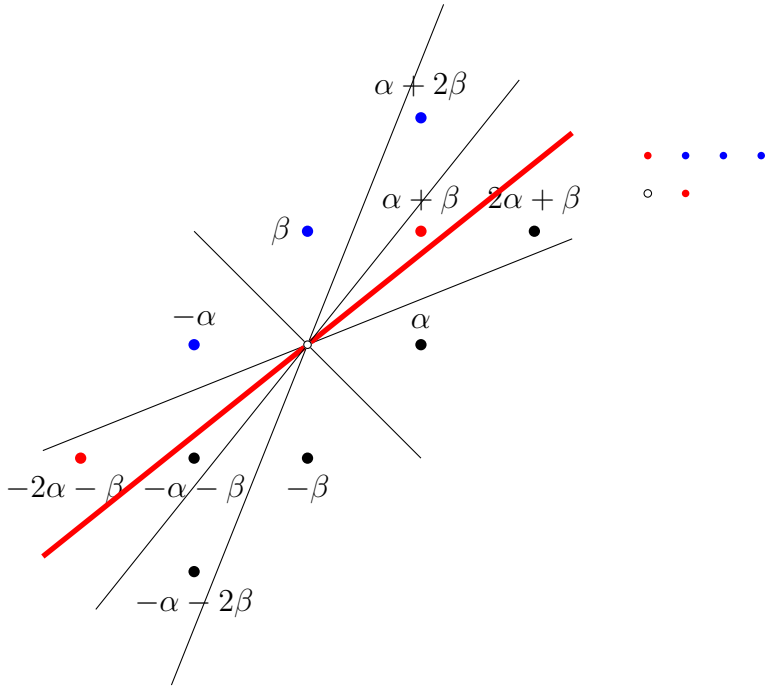
Let $(R, \mathbb{B})$ be a generalized root system and assume that $|R| < \infty$. Such a generalized root system is called a **finite generalized root system**. When $|R| < \infty$, one has $\mathbb{B} = \mathbb{B}_R$. Hence, in this case, we simply write $(R, \mathbb{B})$ as R.

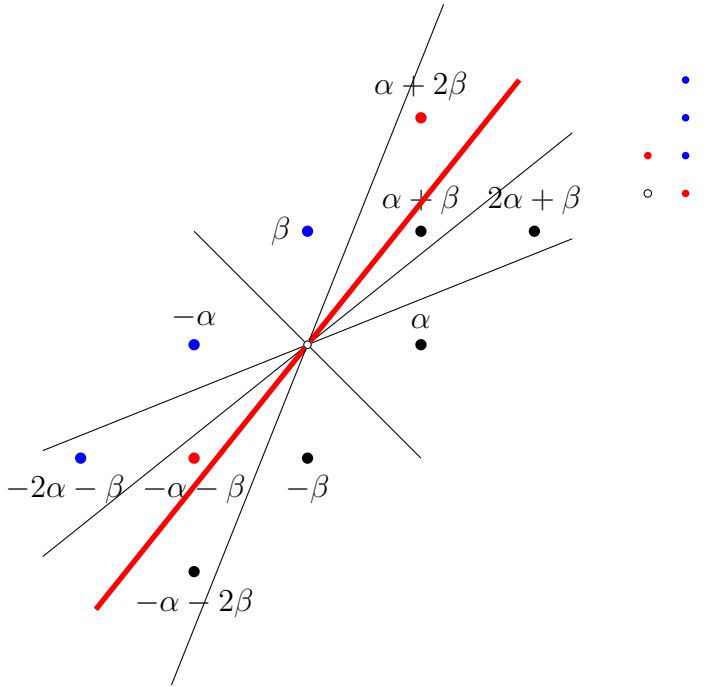
Heckenberger の対角型 Nichols 代数の分類リスト (2010)
階数 2 の分類表の 9 番目にある対角型 Nichols 代数
の有限一般化されたルート系

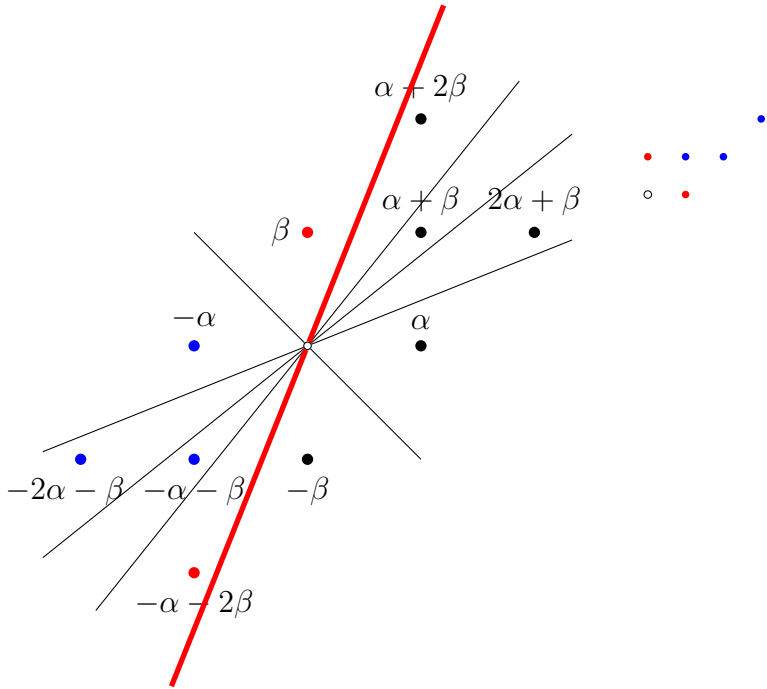


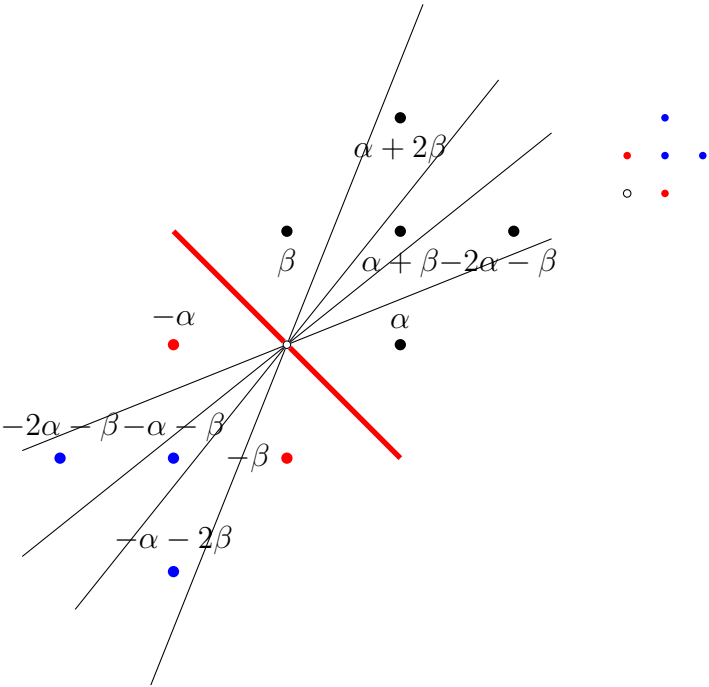

$\alpha + 2\beta$
$\alpha + \beta$
$2\alpha + \beta$
β
$-\alpha$
α
$-2\alpha - \beta$
$-\alpha - \beta$
$-\beta$
$-\alpha - 2\beta$

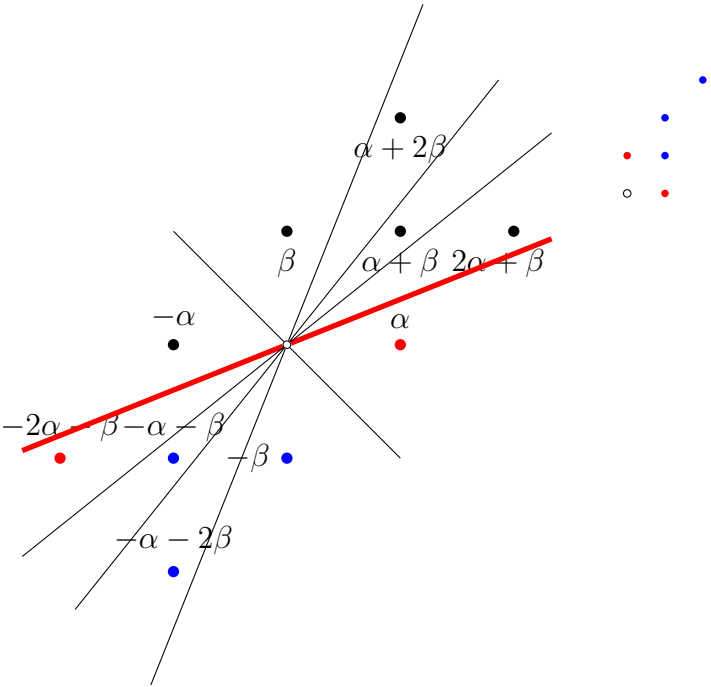

$\alpha + 2\beta$
$\alpha + \beta$
$2\alpha + \beta$
β
$-\alpha$
α
$-2\alpha - \beta$
$-\alpha - \beta$
$-\beta$
$-\alpha - 2\beta$

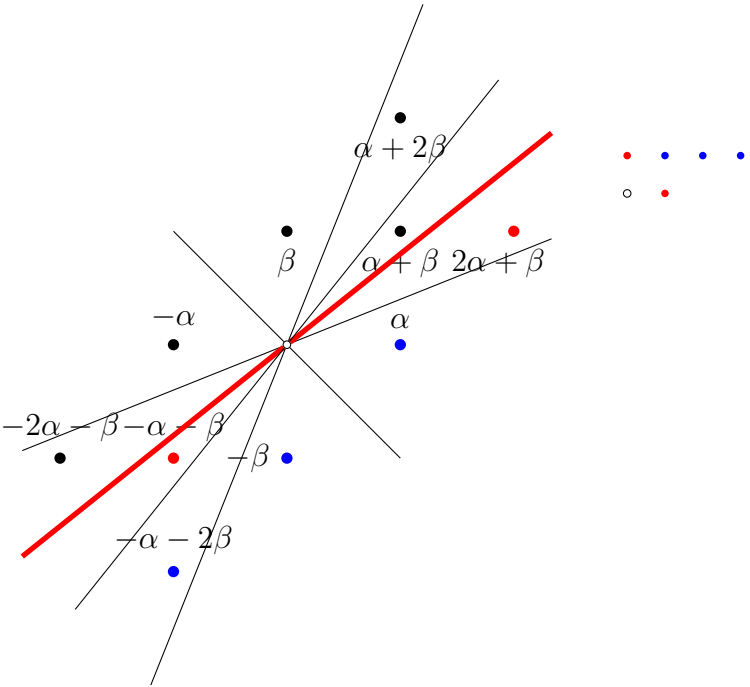

$\alpha + 2\beta$
$\alpha + \beta$
$2\alpha + \beta$
β
$-\alpha$
α
$-2\alpha - \beta$
$-\alpha - \beta$
$-\beta$
$-\alpha - 2\beta$

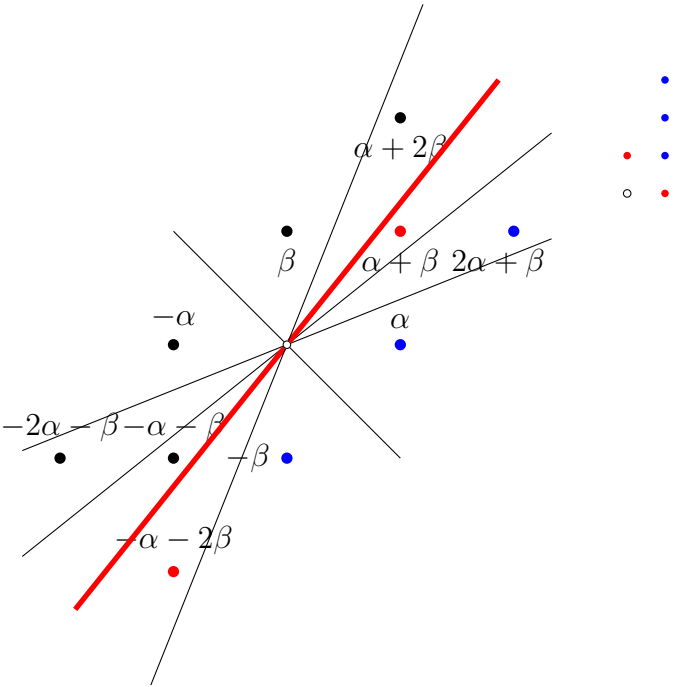

$\alpha + 2\beta$
$\alpha + \beta$
$2\alpha + \beta$
β
$-\alpha$
α
$-2\alpha - \beta$
$-\alpha - \beta$
$-\beta$
$-\alpha - 2\beta$

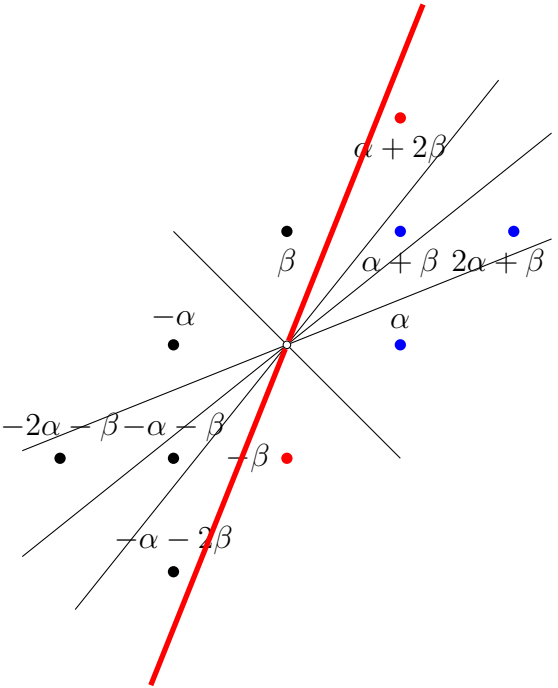

$\alpha + 2\beta$
$\alpha + \beta$
$2\alpha + \beta$
β
$-\alpha$
α
$-2\alpha - \beta$
$-\alpha - \beta$
$-\beta$
$-\alpha - 2\beta$


$\alpha + 2\beta$
β
$\alpha + \beta - 2\alpha - \beta$
$-\alpha$
α
$-2\alpha - \beta - \alpha - \beta$
$-\beta$
$-\alpha - 2\beta$


$\alpha + 2\beta$
β
$\alpha + \beta$
$2\alpha + \beta$
$-\alpha$
α
$-2\alpha - \beta$
$-\alpha - \beta$
$-\beta$
$-\alpha - 2\beta$


$\alpha + 2\beta$
β
$\alpha + \beta$
$2\alpha + \beta$
$-\alpha$
α
$-2\alpha - \beta$
$-\alpha - \beta$
$-\beta$
$-\alpha - 2\beta$


$\alpha + 2\beta$
β
$\alpha + \beta$
$2\alpha + \beta$
$-\alpha$
α
$-2\alpha - \beta$
$-\alpha - \beta$
$-\beta$
$-\alpha - 2\beta$


$\alpha + 2\beta$
β
$\alpha + \beta$
$2\alpha + \beta$
$-\alpha$
α
$-2\alpha - \beta$
$-\alpha - \beta$
$-\beta$
$-\alpha - 2\beta$

Finite generalized root systems were classified by Cuntz-Heckenberger (2015).

For a finite generalized root system $R = (R, \mathbb{B})$, define a graph (V, E) by

$$V = \mathbb{B}$$

(the vertex set) and

$$E = \left\{ \{\Pi, \Pi^{(\alpha)}\} \mid \Pi \in \mathbb{B},\ \alpha \in \Pi \right\}$$

(the edge set).

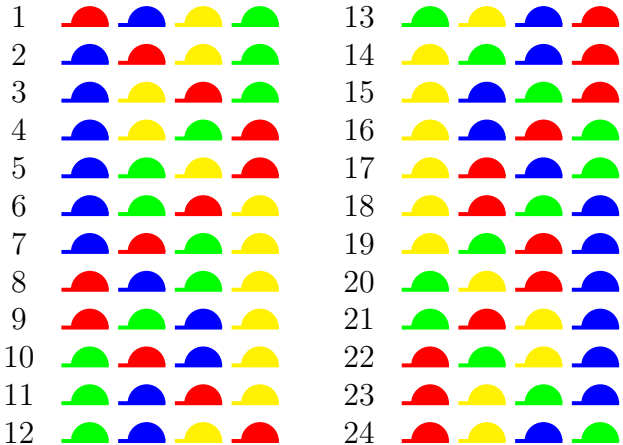
We call (V, E) the Cayley graph of R.

Theorem 1. (Yamane (2022), Inoue–Yamane (2024)) The Cayley graph of every finite generalized root system R admits a Hamiltonian cycle.

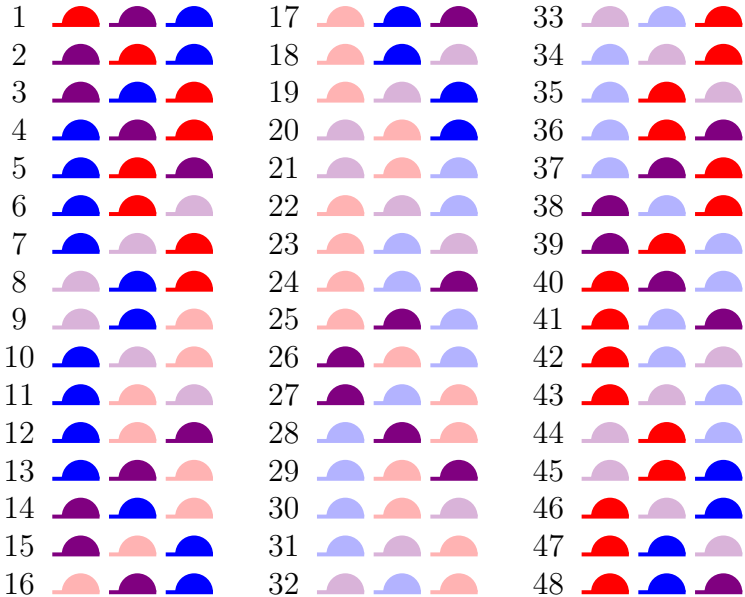
A_{n-1} **型：** $R = \{\varepsilon_i - \varepsilon_j \,|\, i, j \in J_{1,n},\, i \neq j\}$,
$\mathbb{B} = \{\{\alpha_i = \varepsilon_{k_i} - \varepsilon_{k_{i+1}}\ (i \in J_{1,n-1})\} \,|\, \{k_t | t \in J_{1,n}\} = J_{1,n}\}$

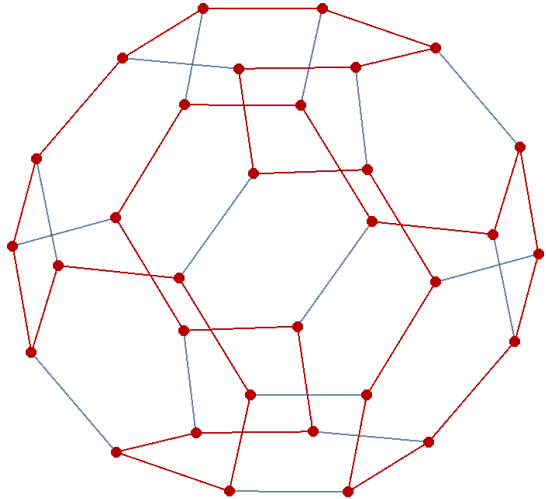
B_n **型：** $R = \{a\varepsilon_i + b\varepsilon_j \,|\, i, j \in J_{1,n},\, i < j,\, a, b \in \{\pm 1\}\} \cup \{a\varepsilon_i \,|\, i \in J_{1,n},\, a \in \{\pm 1\}\}$,
$\mathbb{B} = \{\{\alpha_i = a_i\varepsilon_{k_i} - a_{i+1}\varepsilon_{k_{i+1}}\ (i \in J_{1,n-1}),\, \alpha_n = a_n\varepsilon_{k_n}\} \,|\, \{k_t | t \in J_{1,n}\} = J_{1,n},\, a_t \in \{\pm 1\}\}$

A_3 型 R の Hamilton 閉路を 4 色の体育帽で表現した。

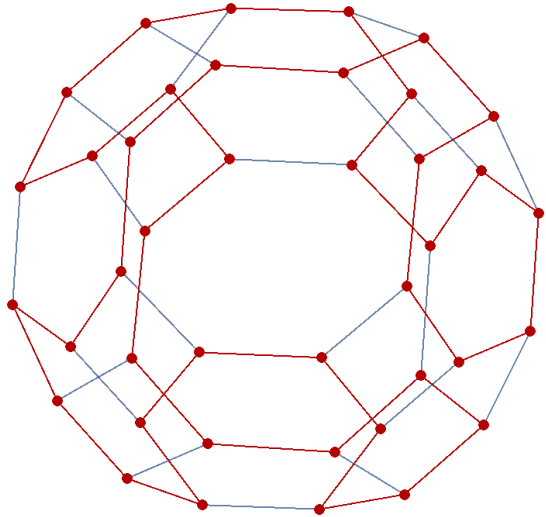


B_3 型 R の Hamilton 閉路を 3 色の体育帽とその裏返しを用いて表現した。

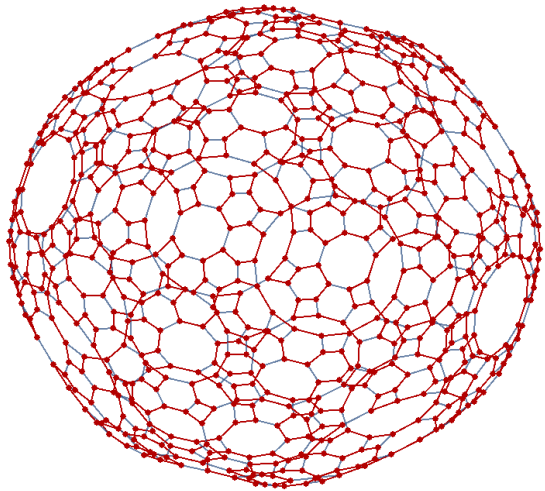




Nr.2



Nr.3



Nr.55

Dijkstra's Algorithm (1956, 1959)

Let $G = (V, E)$ be a finite connected graph. Assume that $|V| \geq 2$.
Let $w : E \to \mathbb{N}$ be a map.
Define a map $M : V \times V \to \mathbb{Z}_{\geq 0}$ by

$$M(u,v) = \begin{cases} 0 \quad (\text{if } u = v), \\ \min\{\sum_{i=1}^{k} w(y_{i-1}, y_i) \mid \exists k \in \mathbb{N}, \exists y_p \in V\,(p \in J_{0,k}), \\ \qquad\qquad y_0 = u, y_k = v, \{y_{p-1}, y_p\} \in E\,(p \in J_{1,k})\} \\ \quad (\text{if } u \neq v). \end{cases}$$

Theorem 2 (Dijkstra's Algorithm (1956, 1959))**.** Let

$$N = \sum_{e \in E} w(e).$$

Let $s, t \in V$ with $s \neq t$. Construct

$$n \in J_{1,|V|-1}, \qquad f_i : V \to \mathbb{Z}_{\geq 0} \quad (i \in J_{0,n}),$$

and

$$y_i \in V \quad (i \in J_{0,n})$$

by the following steps (algorithm).

Then

$$y_0 = s \text{ and } M(s,t) = f_n(t)$$

hold.

(Step 0) Set $y_0 = s$, and define f_0 by

$$f_0(y_0) = 0, \qquad f_0(u) = N + 1 \quad (u \in V \setminus \{y_0\}).$$

Assume that

$$k \in J_{0,n-1}, \qquad t \notin \{y_i \,|\, i \in J_{0,k-1}\}.$$

Let

$$T_k = V \setminus \{y_i \,|\, i \in J_{0,k-1}\}.$$

(Step 1)$_k$ Choose a vertex

$$z \in T_k$$

for which $f_k(z)$ is minimal among all vertices in T_k. Set $y_k = z$.
(When $k = 0$, we have $T_0 = V$.)
Then

$$f_k(y_k) \leq N.$$

(Step 2)$_k$ If $z = t$, set

$$n = k, \qquad y_n = t,$$

and terminate the algorithm.

If $z \neq t$, proceed to (Step 3)$_k$.

(Step 3)$_k$ Define f_{k+1} by

$$f_{k+1}(u) = \begin{cases} f_k(z) + w(u,z) & (u \in T_k \setminus \{z\},\ \{u,z\} \in E, \\ & f_k(u) > f_k(z) + w(u,z)), \\ f_k(u) & (\text{otherwise}). \end{cases}$$

$$(\sharp) \qquad M(s,u) \leq f_{k+1}(u) \leq f_k(u) \qquad (u \in V).$$

(Step 4)$_k$ Proceed to (Step 1)$_{k+1}$.

Historical Remark on Dijkstra's Algorithm

Dijkstra's algorithm is one of the most important algorithms in computer science. It was developed by the Dutch computer scientist Edsger W. Dijkstra in 1956 and was published in 1959. The algorithm was designed to solve the shortest-path problem in a weighted graph, namely, to determine a path of minimum total weight from a given vertex to all other vertices. Since its publication, Dijkstra's algorithm has become a fundamental tool in graph theory, optimization, operations research, and computer science.

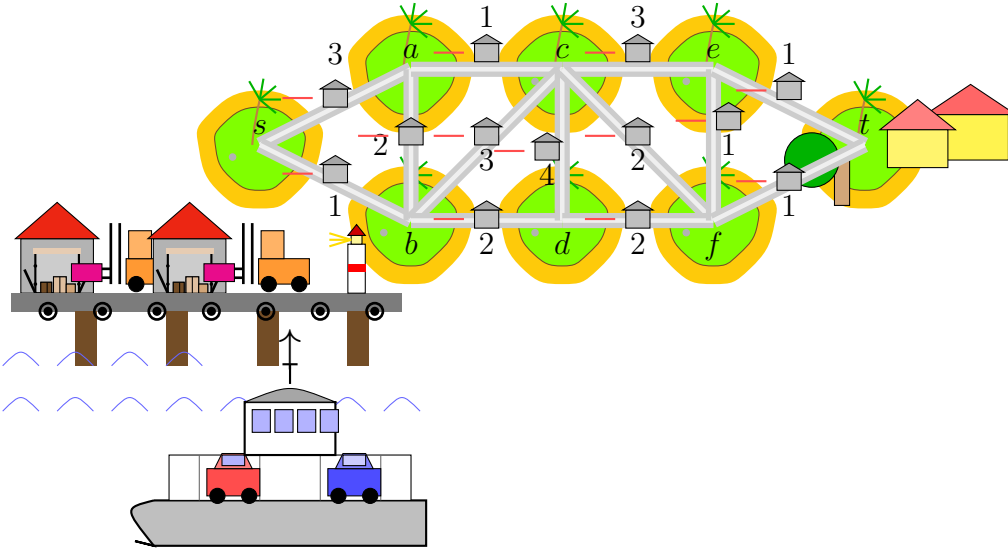
A well-known anecdote is that Dijkstra conceived the main idea of the algorithm in a short time while sitting at a cafe with his fiancee. Its simplicity, elegance, and efficiency have made it one of the most widely used algorithms in modern technology.

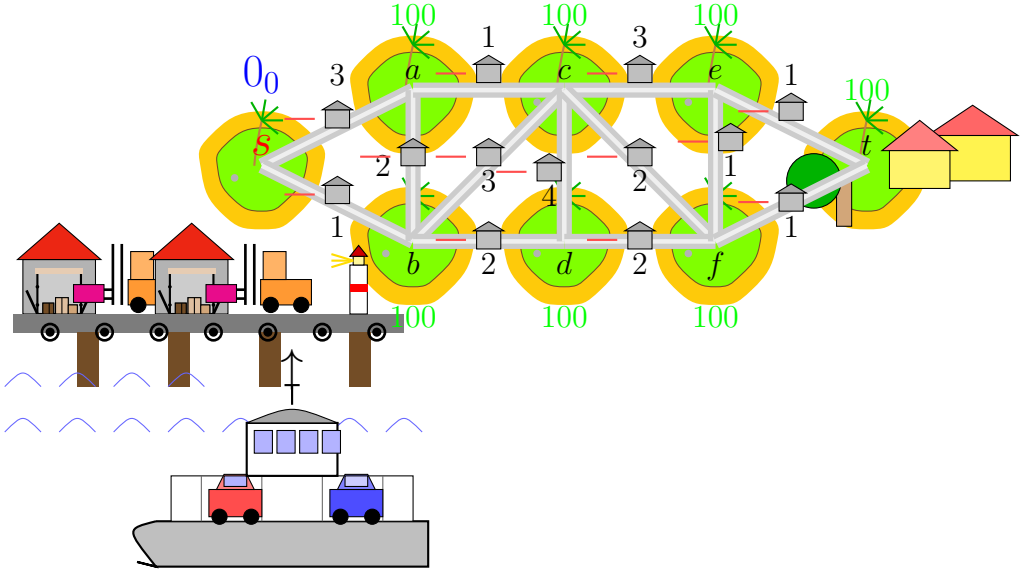
Applications in Everyday Life: Dijkstra's algorithm plays an important role in many aspects of our daily lives. One of its most familiar applications is route navigation. GPS devices and online map services use shortest-path algorithms to determine efficient routes between locations.

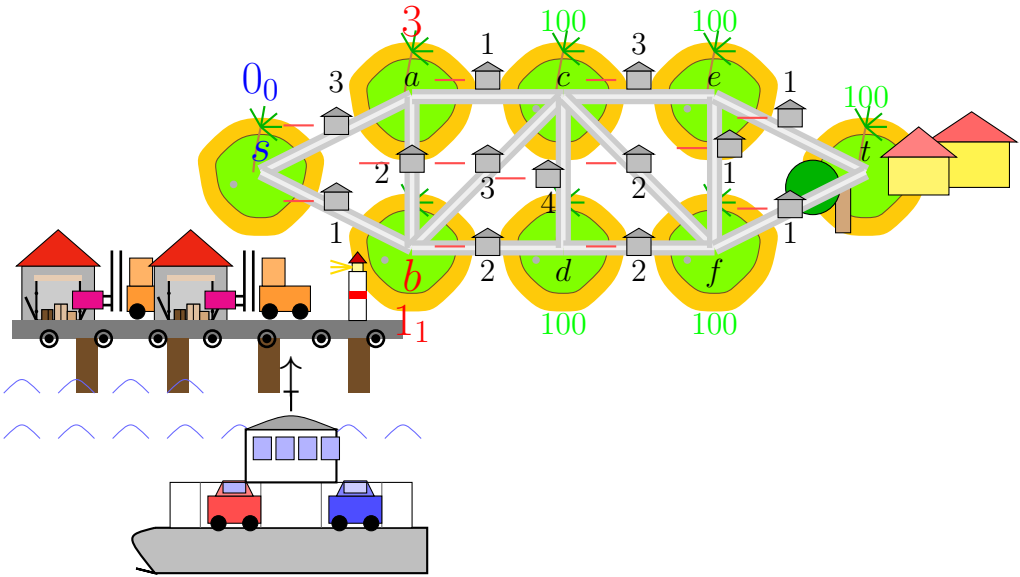
The algorithm is also used in communication networks, where data packets must be transmitted through efficient routes on the Internet. Further applications include transportation planning, airline scheduling, logistics, robotics, and computer games. Whenever a problem can be represented as finding an efficient path in a network, Dijkstra's algorithm or one of its variants is often employed.

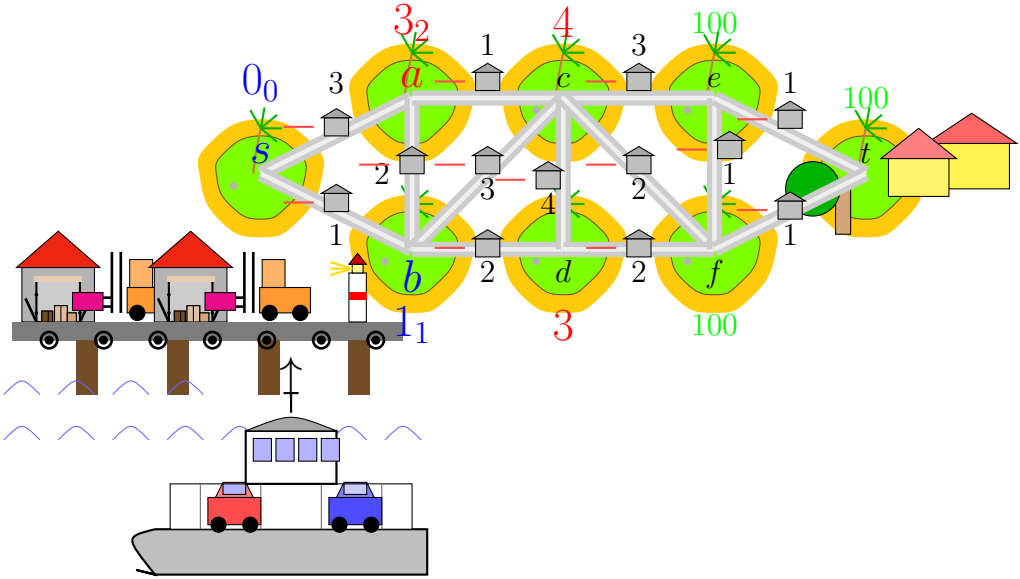
Thus, although it was invented more than sixty years ago, Dijkstra's algorithm remains an indispensable component of modern information technology and continues to influence both theoretical research and practical applications.

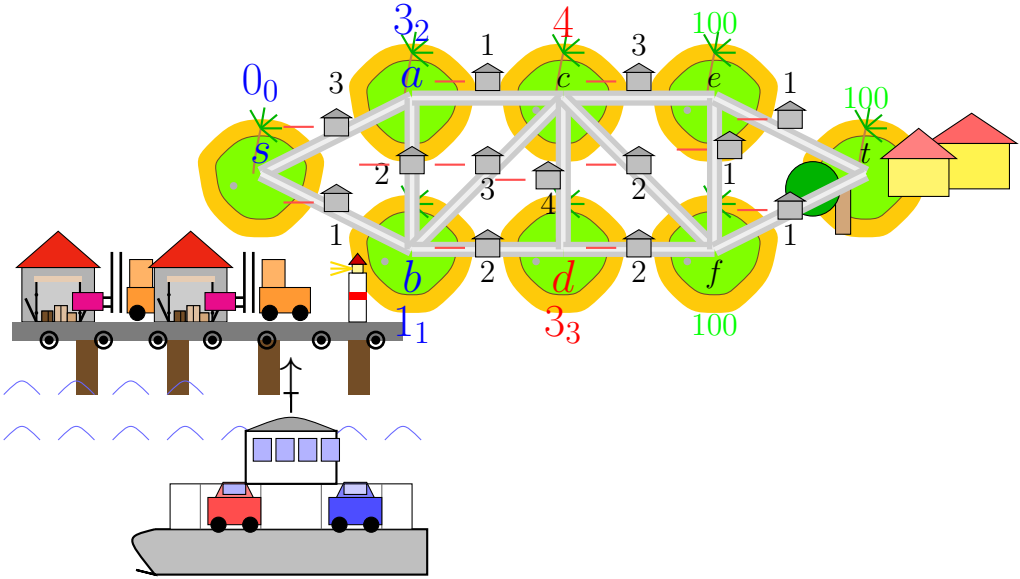
Find the least expensive route from the port to the town.

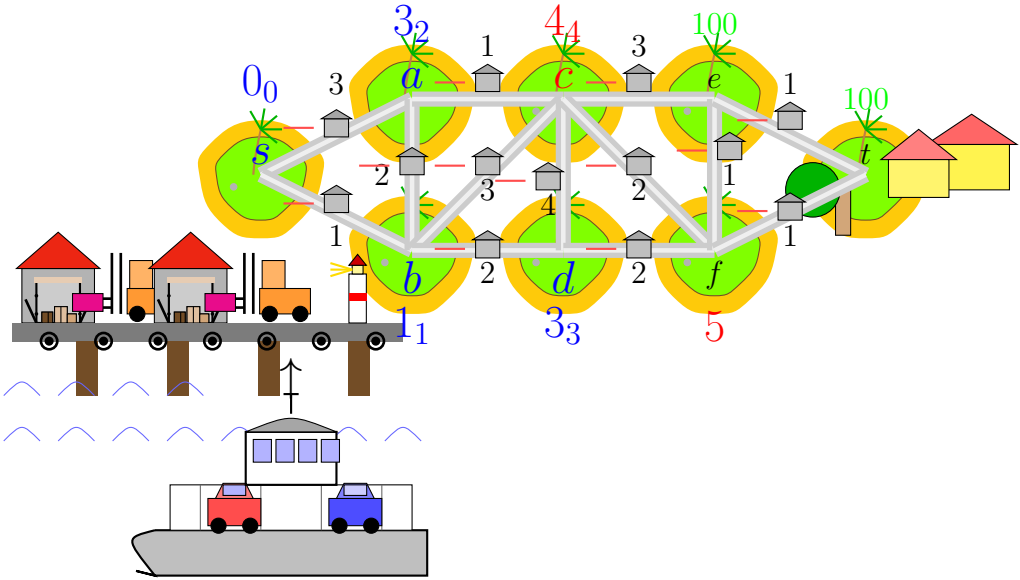


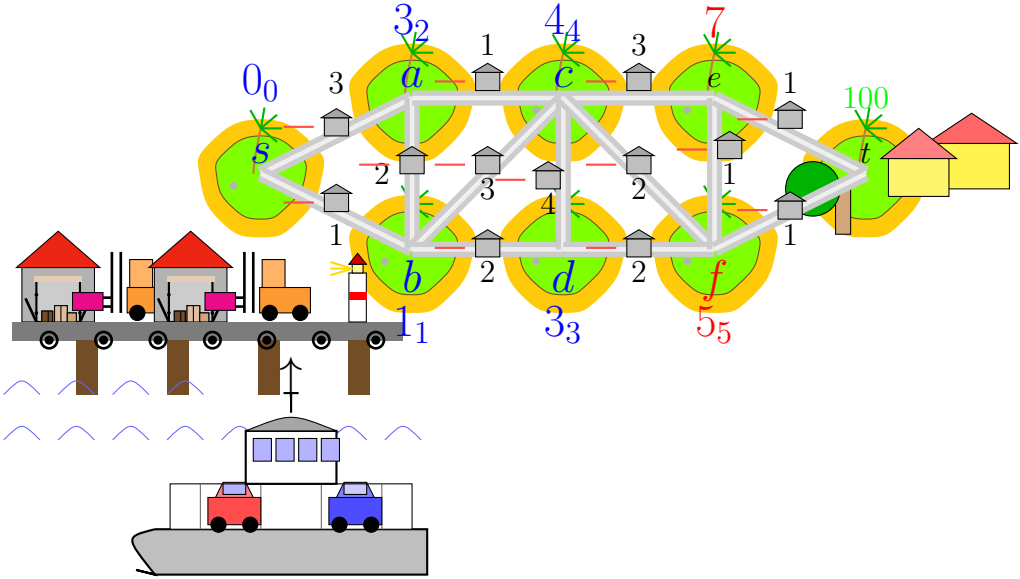

100
100
100
0_0
3
1
3
a
c
e
1
100
s
t
2
3
4
2
1
1
1
b
2
d
2
f
100
100
100


3
100
100
1
3
0_0
3
a
c
e
1
100
s
t
2
3
4
2
1
1
1
b
d
f
2
2
1_1
100
100

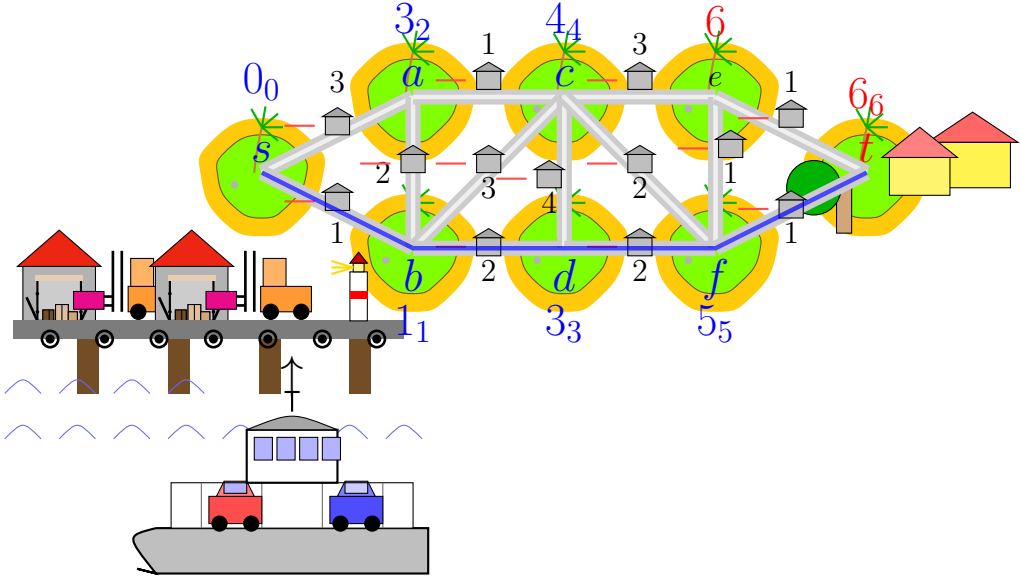

0_0
3_2
4
100
100
100
s
a
c
e
t
b
d
f
1_1
3
3
1
3
1
2
3
4
2
1
1
1
2
2


3_2
4
100
0_0
3
a
1
c
3
e
1
100
s
2
3
2
1
4
t
1
1
2
2
1
b
d
f
1_1
3_3
100


0_0
3_2
4_4
100
100
s
a
c
e
t
b
d
f
1_1
3_3
5
3
1
3
1
2
3
4
2
1
1
2
2
1


0_0
3_2
4_4
7
100
s
a
c
e
t
b
d
f
1_1
3_3
5_5
3
1
2
3
4
2
1
1
2
2
1
1

$s \to b \to d \to f \to t$ the least expensive 6Yen 円 route.



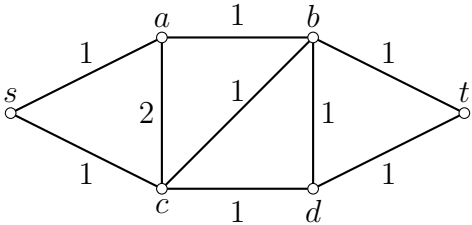
グラフ (V, E) を $|V| = 6$ であるものとし，頂点集合 V，辺集合 E をそれぞれ

$$V = \{s, a, b, c, d, t\}$$

$$E = \{\{s, a\},\ \{s, c\},\ \{a, b\},\ \{a, c\},\ \{b, c\},\ \{b, d\},\ \{b, t\},\ \{c, d\},\ \{d, t\}\}$$

により定める。

$w(\{s, a\}) = 1$, $w(\{s, c\}) = 1$, $w(\{a, b\}) = 1$, $w(\{a, c\}) = 2$, $w(\{b, c\}) = 1$, $w(\{b, d\}) = 1$, $w(\{b, t\}) = 1$, $w(\{c, d\}) = 1$, $w(\{d, t\}) = 1$, $N = 11$



	y_0	y_1	y_3	y_2	y_4	y_5
	s	a	b	c	d	t
f_0	$\hat{0}$	11	11	11	11	11
f_1	$\hat{0}$	$\hat{1}$	11	1	11	11
f_2	0	$\hat{1}$	2	$\hat{1}$	11	11
f_3	0	1	$\hat{2}$	$\hat{1}$	2	11
f_4	0	1	$\hat{2}$	1	$\hat{2}$	3
f_5	0	1	2	1	$\hat{2}$	$\hat{3}$

$T_0 = \{s, a, b, c, d, t\}$, $T_1 = \{a, b, c, d, t\}$, $T_2 = \{b, c, d, t\}$, $T_3 = \{b, d, t\}$,
$T_4 = \{d, t\}$, $T_5 = \{t\}$
$M(s, t) = 3$

最少重み経路を求める逆経路は

$(f_5, y_5) \leftarrow (f_4, y_5) \leftarrow (f_4, y_3) \leftarrow (f_3, y_3) \leftarrow (f_2, y_3) \leftarrow (f_2, y_1) \leftarrow (f_1, y_1)$
$\leftarrow (f_1, y_0) \leftarrow (f_0, y_0)$

最少重み経路を与える経路は

$y_0 \rightarrow y_1 \rightarrow y_3 \rightarrow y_5 = s \rightarrow a \rightarrow b \rightarrow t$

Ford-Fulkerson's Algorithm (1956)

Let $G = (V, E)$ be a finite connected graph. Assume that $|V| \geq 2$. Let $c : E \to \mathbb{R}_{>0}$ be a map. Let $s, t \in V$ satisfy $s \neq t$. The vertex s is called the **source**, and the vertex t is called the **sink**.

Let $\mu : V \times V \to J_{-1,1}$ be a map satisfying (μ1)-(μ4) below.

(μ1) $\mu(u, v) + \mu(v, u) = 0 \qquad (u, v \in V)$
(μ2) $\mu(u, v) \neq 0$ if and only if $\{u, v\} \in E$
(μ3) $\mu(s, v) = 1 \qquad (\{s, v\} \in E)$
(μ4) $\mu(u, t) = 1 \qquad (\{u, t\} \in E)$

A map $f : E \to \mathbb{R}_{\geq 0}$ is called a **flow** of (V, E, s, t, μ, c) if it satisfies (fl1) and (fl2) below.

(fl1) For every $\{u, v\} \in E$,

$$f(\{u, v\}) \leq c(\{u, v\}).$$

(fl2) For every $u \in V \setminus \{s, t\}$,

$$\sum_{\{u,v\} \in E} f(\{u, v\})\mu(u, v) = 0.$$

Let $\mathcal{F}$ denote the set of all flows of (V, E, s, t, μ, c).

For $f \in \mathcal{F}$, define

$$\text{vol}(f) = \sum_{v \in V,\ \{s,v\} \in E} f(\{s, v\}).$$

Then

$$\mathrm{vol}(f) = \sum_{u \in V,\ \{u,t\} \in E} f(\{u,t\})$$

holds.

Define

$$\mathcal{M}(\mathcal{F}) = \max\{\mathrm{vol}(f) \mid f \in \mathcal{F}\}.$$

The quantity $\mathcal{M}(\mathcal{F})$ is called the **maximum flow** of $\mathcal{F}$.

Theorem 3. (Ford-Fulkerson's algorithm (1956))

Assume that $c(E) \subset \mathbb{N}$. Let $n \in \mathbb{Z}_{\geq 0}$ and $f_i \in \mathcal{F}$ $(i \in J_{0,n})$ be constructed by the following Steps (algorithm). Then

$$\mathcal{M}(\mathcal{F}) = \mathrm{vol}(f_n).$$

(Step 0) Define $f_0 \in \mathcal{F}$ by

$$f_0(\{u, v\}) = 0 \qquad (\{u, v\} \in E).$$

(Step 1)$_0$ Let $m_0 \in \mathbb{N}$ and let

$$p_0 : J_{0,m_0} \to V$$

be an injective map satisfying

$$p_0(0) = s, \quad p_0(m_0) = t,$$
$$\{p_0(k), p_0(k+1)\} \in E, \quad \mu(p_0(k), p_0(k+1)) = 1 \quad (k \in J_{0,m_0-1}).$$

(If no such p_0 exists, then the algorithm terminates at (Step 0), and $\mathcal{M}(\mathcal{F}) = 0$ and $n = 0$.)

(Step 2)$_0$ Define

$$\tau_{f_0}(p_0) = \min\{c(\{p_0(k), p_0(k+1)\}) \,|\, k \in J_{0,m_0-1}\}.$$

(Step 3)$_0$ Define $f_1 \in \mathcal{F}$ by

$$f_1(\{u,v\}) = \begin{cases} \tau_{f_0}(p_0) & \text{if } \{u,v\} = \{p_0(k), p_0(k+1)\} \text{ for some } k \in J_{0,m_0-1}, \\ 0 & \text{otherwise.} \end{cases}$$

for all $\{u, v\} \in E$.

Let $x \in \mathbb{N}$ with $x \geq 1$.

(Step 1)$_x$ Let $m_x \in \mathbb{N}$ and let

$$p_x : J_{0,m_x} \to V$$

be an injective map satisfying

$$p_x(0) = s, \quad p_x(m_x) = t, \quad \{p_x(k), p_x(k+1)\} \in E \quad (k \in J_{0,m_x-1}),$$

and

for each $k \in J_{0,m_x-1}$, either $(\alpha 1)$ or $(\alpha 2)$ holds:

$(\alpha 1)$ $\mu(p_x(k), p_x(k+1)) = 1$ and $f_x(\{p_x(k), p_x(k+1)\}) < c(\{p_x(k), p_x(k+1)\}$

$(\alpha 2)$ $\mu(p_x(k), p_x(k+1)) = -1$ and $0 < f_x(\{p_x(k), p_x(k+1)\})$.

(If no such p_x exists, then the algorithm terminates at (Step 3)$_{x-1}$, and

$$\mathcal{M}(\mathcal{F}) = \mathrm{vol}(f_x), \qquad n = x.$$

)

(Step 2)$_x$

Define

$$\begin{aligned}
\tau'_{f_x}(p_x) &= \min\{c(\{p_x(k), p_x(k+1)\}) - f_x(\{p_x(k), p_x(k+1)\}) \mid \\
&\quad k \in J_{0,m_x-1},\ \mu(p_x(k), p_x(k+1)) = 1\}, \\
\tau''_{f_x}(p_x) &= \min\{f_x(\{p_x(k), p_x(k+1)\}) \mid \\
&\quad k \in J_{0,m_x-1},\ \mu(p_x(k), p_x(k+1)) = -1\}.
\end{aligned}$$

Define

$$\tau_{f_x}(p_x) = \min\{\tau'_{f_x}(p_x), \tau''_{f_x}(p_x)\}\ (> 0).$$

(Step 3)$_x$ Define $f_{x+1} \in \mathcal{F}$ by

$$f_{x+1}(\{u,v\}) =$$

$$\begin{cases} f_x(\{u,v\}) + \tau_{f_x}(p_x) & \text{if there exists } k \in J_{0,m_x-1} \text{ such that} \\ \qquad \{u,v\} = \{p_x(k), p_x(k+1)\} \text{ and } \mu(p_x(k), p_x(k+1)) = 1, \\ f_x(\{u,v\}) - \tau_{f_x}(p_x) & \text{if there exists } k \in J_{0,m_x-1} \text{ such that} \\ \qquad \{u,v\} = \{p_x(k), p_x(k+1)\} \text{ and } \mu(p_x(k), p_x(k+1)) = -1, \\ 0 \quad \text{otherwise.} \end{cases}$$

for all $\{u,v\} \in E$.

Historical Remark on Ford-Fulkerson's Algorithm

The Ford-Fulkerson algorithm was introduced by L. R. Ford, Jr. and D. R. Fulkerson in 1956 in their study of transportation and communication networks. It was the first systematic method for solving the *maximum-flow problem*, namely, the problem of determining the largest amount of material that can be sent from a source to a sink through a network with capacity constraints.

The central idea of the algorithm is to start with the zero flow and to increase the flow step by step along suitable paths from the source to the sink. By repeatedly augmenting the current flow, one eventually obtains a maximum flow. This simple but powerful idea became one of the fundamental concepts of combinatorial optimization and graph

theory.

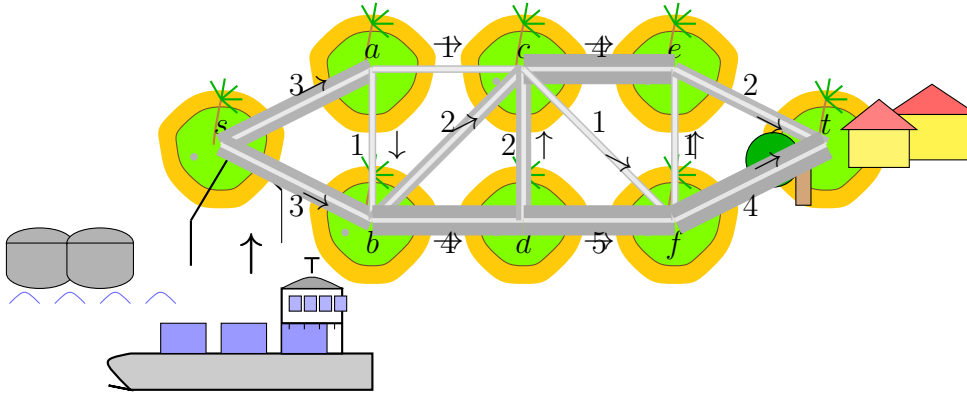
The work of Ford and Fulkerson also led to the celebrated *Max-Flow Min-Cut Theorem*, which states that the value of a maximum flow is equal to the capacity of a minimum cut. This theorem is regarded as one of the most important results in discrete mathematics and optimization theory.

Applications in Everyday Life

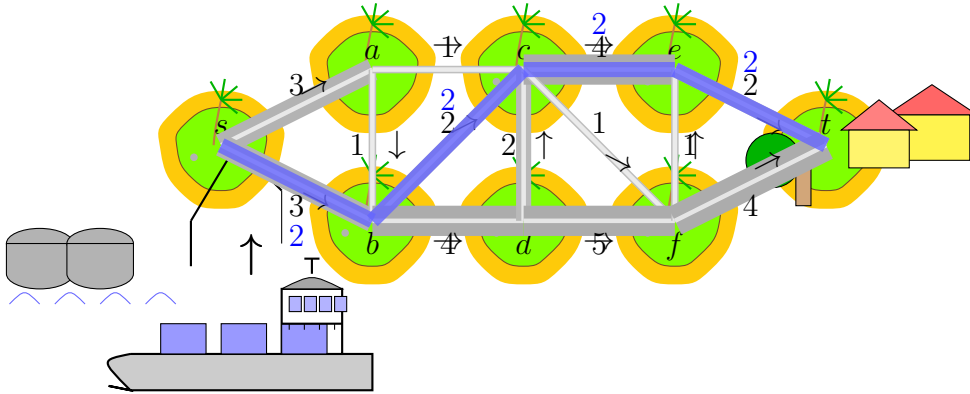
Ford-Fulkerson's algorithm and its variants have numerous practical applications. They are used in transportation networks to maximize the movement of goods and passengers, in water-supply systems to determine the greatest possible distribution of water, and in communication networks to maximize data transmission.

The algorithm is also employed in logistics, traffic engineering, airline scheduling, project planning, and computer vision. Modern Internet infrastructure relies heavily on techniques derived from network-flow theory.

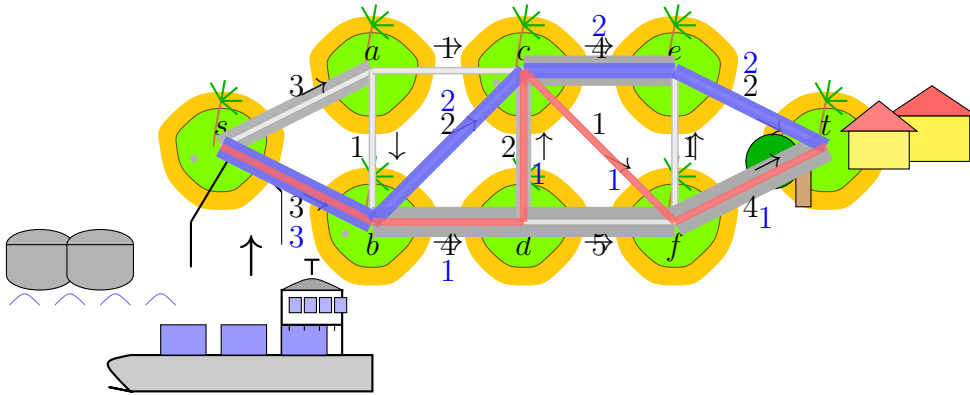
Thus, the Ford-Fulkerson algorithm is not merely a theoretical result in graph theory; it is an essential tool for solving many real-world optimization problems involving the efficient use of limited resources.


a
c
e
s
t
b
d
f
1
4
3
2
2
1
1
1
3
4
5

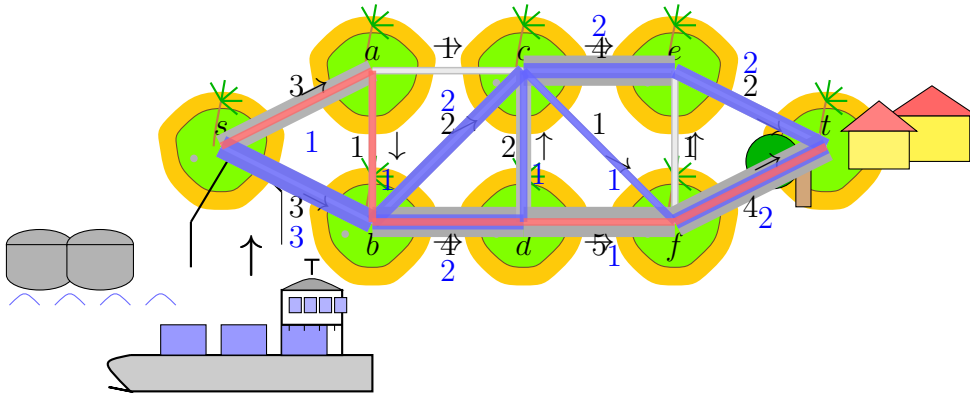
Send 2 tons of water.



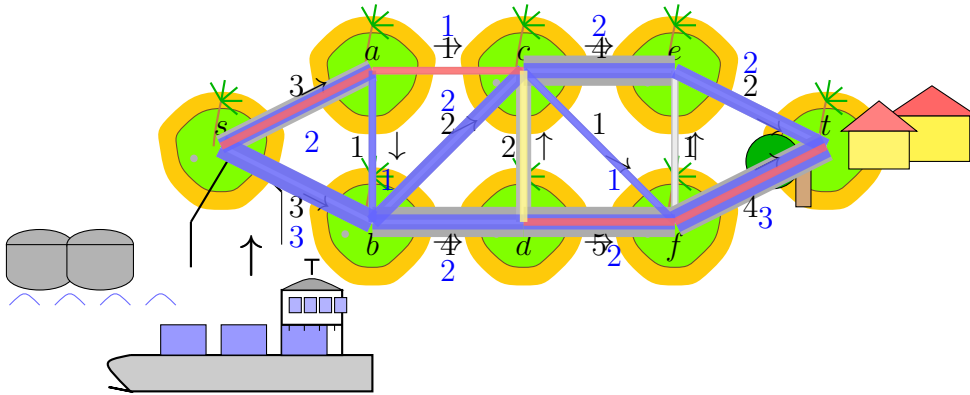
Send 1 additional ton along another path,
yielding a total flow of 3 tons.



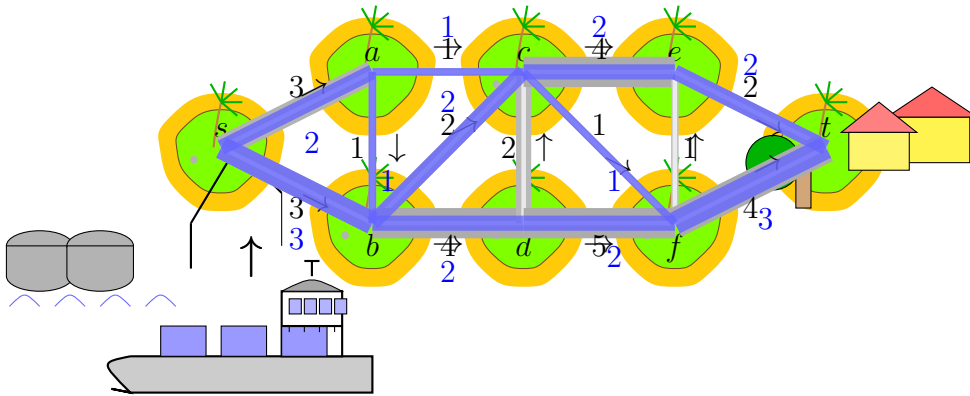
Send 1 more ton of water along another path,
making a total of 4 tons.



Although some pipes carry less water,
1 more ton can be sent along another path,
giving a total flow of 5 tons.



Hence, the maximum flow is 5.



例-FF

$V = \{s, a, b, c, d, t\}$,

$E = \{\{s, a\}, \{s, b\}, \{a, b\}, \{a, c\}, \{a, d\}, \{b, d\}, \{c, d\}, \{c, t\}, \{d, t\}\}$,

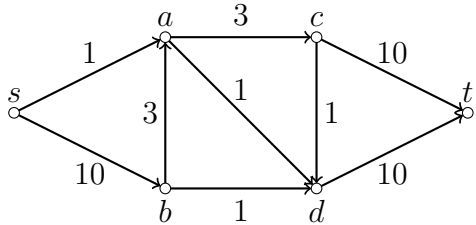
$\mu(s, a) = 1,\ \mu(s, b) = 1,\ \mu(a, b) = -1,\ \mu(a, c) = 1,\ \mu(a, d) = 1,$
$\mu(b, d) = 1,\ \mu(c, d) = 1,\ \mu(c, t) = 1,\ \mu(d, t) = 1,$

$c(\{s, a\}) = 1,\ c(\{s, b\}) = 10,\ c(\{a, b\}) = 3,\ c(\{a, c\}) = 3,\ c(\{a, d\}) =$
$1,\ c(\{b, d\}) = 1,\ c(\{c, d\}) = 1,\ c(\{c, t\}) = 10,\ c(\{d, t\}) = 10,$

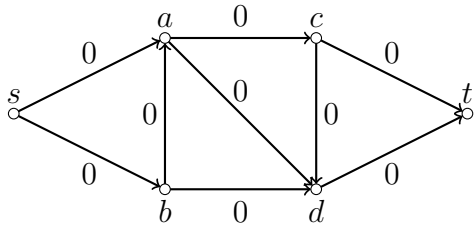
	$\{s,a\}$	$\{s,b\}$	$\{a,b\}$	$\{a,c\}$	$\{a,d\}$	$\{b,d\}$	$\{c,d\}$	$\{c,t\}$	$\{d,t\}$
μ	1	1	-1	1	1	1	1	1	1
c	1	10	3	3	1	1	1	10	10
f_0	0	0	0	0	0	0	0	0	0
f_1	1	0	0	1	0	0	0	1	0
f_2	1	1	1	2	0	0	1	1	1
f_3	1	2	1	2	0	1	0	2	1
f_4	1	3	2	2	1	1	0	2	2
f_5	1	4	3	3	1	1	1	2	3

$n = 5$, $\mathcal{M}(\mathcal{F}) = 5$ ($\because$ S, $T = V \setminus S$ を定理 ??の証明のものとすると $S = \{s, a, b\}$ であるので $t \notin T$ より $p_5(\cdot)$ を見つけることが出来ない。)

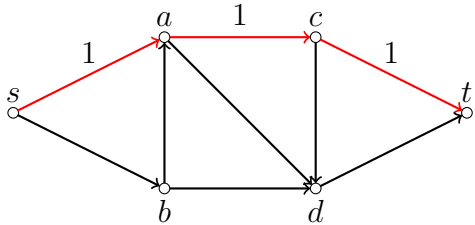
$c(\cdot),\ \mu(\cdot)$



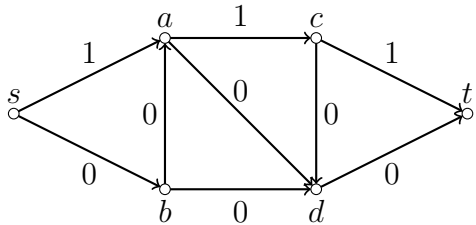
$f_0(\cdot)$



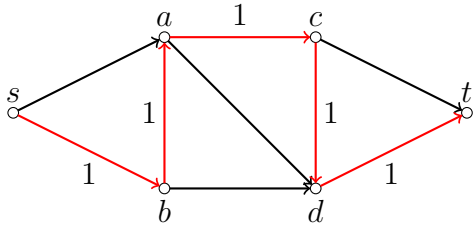
$p_0(\cdot)$, $\tau_{f_0}(p_0)$



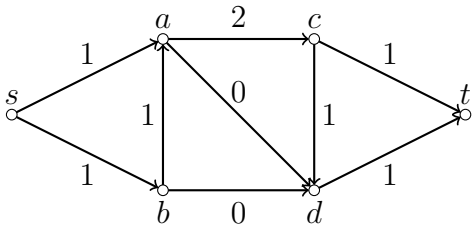
$f_1(\cdot)$



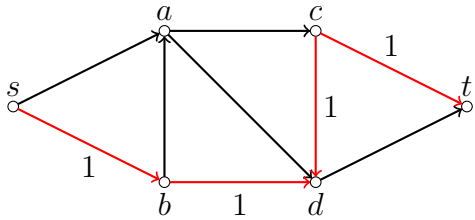
$p_1(\cdot)$, $\tau_{f_1}(p_1)$



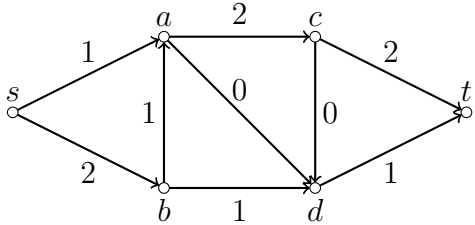
$f_2(\cdot)$



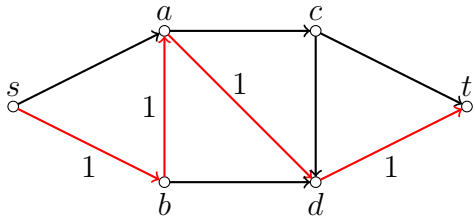
$p_2(\cdot)$, $\tau_{f_2}(p_2)$



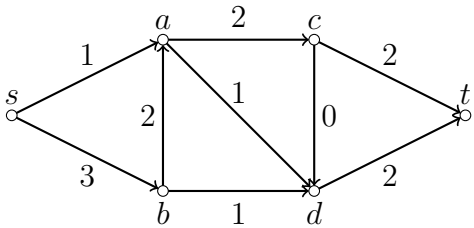
$f_3(\cdot)$



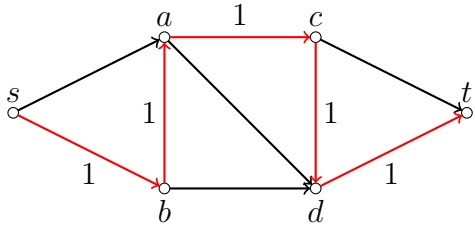
$p_3(\cdot)$, $\tau_{f_2}(p_3)$



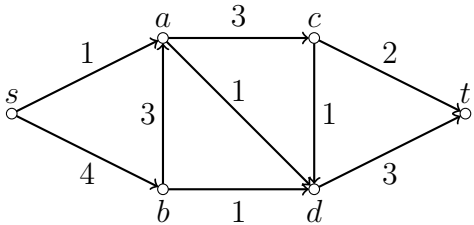
$f_4(\cdot)$



$p_4(\cdot)$, $\tau_{f_4}(p_4)$



$f_5(\cdot)$



ご清聴ありがとうございます。

論文を書きました。

[1] Hiroyuki Yamane: Hamilton circuits of Cayley graphs of Weyl groupoids of generalized quantum groups, Toyama. Math. J. 43 (2022), 1-76

[2] Takato Inoue, Hiroyuki Yamane, Hamiltonian Cycles for Finite Weyl Groupoids, J. Algebra Appl. 25 (2026), no. 7, 2650054, 40 pp.

群

Definition 4. (1) 空でない集合 G が群であるとは次の積と呼ばれる演算が与えられていてと公理 $(G1)$-$(G3)$ を満たすときに言う。(e を e_G とも書く。)

積, $\cdot : G \times G \to G,\ (a, b) \mapsto ab$

$(G1)$ $\forall a, \forall b, \forall c \in G,\ (ab)c = a(bc)$ (結合法則)
$(G2)$ $\exists e \in G,\ \forall a \in G,\ ae = ea = a$ (単位元 e の存在)
$(G3)$ $\forall a \in G,\ \exists a^{-1} \in G,\ aa^{-1} = a^{-1}a = e$ (逆元 a^{-1} の存在)

$|G|$ を G の位数という。$|G| < \infty$ のとき G を有限群という。

Theorem 5.（ラグランジュの定理） G を有限群とし，H を G の部分群とする。$n := |G| (\in \mathbb{N})$, $m := |H| (\in \mathbb{N})$ とする。このとき $m|n$ が成り立つ。とくに $a \in G$ に対して $r := |\langle a \rangle|$ とおくと $r|n$ が成り立ち，$a^n = e$ が成り立つ。さらに，n が素数ならば（素数であるので $n \geq 2$），$b \in G \setminus \{e\}$ に対して $G = \langle b \rangle$ が成り立つ。

Definition 6. （右分解，左分解） G を群とし，H を G の部分群とする。G 上の同値関係 $\equiv_r$ を定義する。$a, b \in G$ とする。

$$a \equiv_r b \quad \overset{\text{def}}{\iff} \quad ab^{-1} \in H$$

G の $\equiv_r$ の同値類の集合 $G/\equiv_r$ を $G\backslash H$ と書く。

$$G\backslash H = \{Ha \,|\, a \in G\}$$

である。$\equiv_r$ が同値関係であるので，選択公理より，G の部分集合 $\{a_i \,|\, i \in$

$I\}$ が存在して（I は添え字集合），次を満たす。

$$G = \bigcup_{i \in I} Ha_i, \quad Ha_i \cap Ha_j = \emptyset \ (i, j \in I, \ i \neq j)$$

このとき，

$$G = \coprod_{i \in I} Ha_i$$

と書き，これを G の H による右分解という。

G 上の同値関係 $\equiv_l$ を定義する。$a, b \in G$ とする。

$$a \equiv_l b \quad \overset{\text{def}}{\iff} \quad a^{-1}b \in H$$

G の $\equiv_l$ の同値類の集合 $G/\equiv_l$ を G/H と書く。

$$G/H = \{aH \,|\, a \in G\}$$

である。G の H による左分解

$$G = \coprod_{i \in I} b_i H$$

を右分解と同様にして定義する。

$G = \coprod_{i \in I} H a_i$（右分解）ならば $G = \coprod_{i \in I} a_i^{-1} H$（左分解）である。