1．序論

ネットワーク，グラフ，アルゴリズム，計算量

2．計算モデル

チューリング機械，多項式時間アルゴリズム，ＮＰ－完全

3．アルゴリズムの手法

データ構造，再帰，動的計画法，divide-and-conquer

4．最小木

5．最短路

6．最大フロー

7．応用

ＶＬＳＩ配線

通信網制御・構成

計算量？

例　巾乗

$30^{(万円)} \times (1+0.06)^{20}$

アルゴリズムＡ	乗算
$30 \times 1.06 \times 1.06 \times \cdots \times 1.06$	２０回
アルゴリズムＢ	
$a = 1.06$	
$a^2 = a \times a$, $a^4 = a^2 \times a^2$	
$a^8 = a^4 \times a^4$, $a^{16} = a^8 \times a^8$	
$30 \times a^{20} = a^{16} \times a^4 \times 30$	６回

$20 = 10100$　（２進）　　基本操作（演算）

時間計算量

領域計算量

X^n

(A')　$x \times x \times x \times \cdots \times x$　　　　n − 1回

(B')　$n = d_k d_{k-1} \cdots d_1 d_0$,　　　$k = [\log_2 n]$

$$\left.\begin{array}{l} x^2 = x \times x \\ x^4 = x^2 \times x^2 \\ \quad\vdots \\ x^{2^k} = x^{2^{k-1}} \times x^{2^{k-1}} \end{array}\right\} k\text{回}$$

$$x^n = \underbrace{x^{d_k \cdot 2^k} \times x^{d_{k-1} \cdot 2^{k-1}} \times \cdots \times x^{d_1 \cdot 2^1} \times x^{d_0 \cdot 2^0}}_{\text{高々}k\text{回}}$$

合計　高々　$2\log_2 n$　回

M(n) :　x^nの計算に必要かつ十分な乗算回数

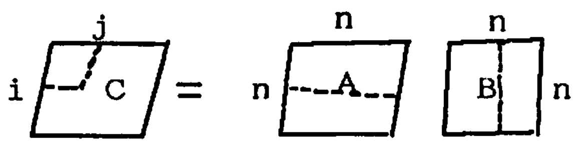
$$\underset{\text{下界}}{\log_2 n} \leq M(n) \leq \underset{\text{上界}}{2\log_2 n}$$

k回の乗算で求まる最大数　x^{2^k}

$$x^n \leq x^{2^k}$$

$$\log n \leq k$$

行列積



$C = A \cdot B$　　　　　$O(n^3)$

$$C_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$　　　　$n^2 \cdot n = n^3$

Strassen

$$n \quad \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \qquad \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

$m_1 = (a_{12} - a_{22}) \cdot (b_{21} + b_{22})$

$m_2 = (a_{11} + a_{22}) \cdot (b_{11} + b_{22})$

$m_3 = (a_{11} - a_{21}) \cdot (b_{11} + b_{12})$

$m_4 = (a_{11} + a_{12}) \cdot b_{22}$

$m_5 = a_{11} \cdot (b_{12} - b_{22})$ $\qquad n^{2.7}$

$m_6 = a_{22} \cdot (b_{21} - b_{11})$ $\qquad n^{2.48}$

$m_7 = (a_{21} + a_{22}) \cdot b_{11}$

$c_{11} = m_1 + m_2 - m_4 + m_6$ $\qquad$ 乗算７回

$c_{12} = m_4 + m_5$ $\qquad$ 加減算１８回

$c_{21} = m_6 + m_7$ $\qquad$ log７≒２．８

$c_{22} = m_2 - m_3 + m_5 - m_7$

$T(n) = 7T(\frac{n}{2}) + 18(\frac{n}{2})^2$ $\qquad T(n) = O(n)^{\log 7}$

$n = 2^k$, $k = \log_2 n$ とする

$$T(n) = 7T(\frac{n}{2}) + 18(\frac{n}{2})^2$$

$$= 7\{7T(\frac{n}{2^2}) + 18(\frac{n}{2^2})^2 + 18(\frac{n}{2})^2$$

$$= 7^2 T(\frac{n}{2^2}) + 7 \cdot 18 \cdot (\frac{n}{\ })^2 + 18(\frac{n}{2})^2$$

- - - - - - - - - - ....

$$= 7^k T(\frac{n}{2^k})$$

$$+ 7^{k-1} \cdot 18 \cdot (\frac{n}{2^k})^2 + 7 \cdot 18 \cdot (\frac{n}{2^2})^2 + 18(\frac{n}{2})$$

$$= 7^k T(1)$$

$$+ \frac{18}{4} n^2 \{1 + \frac{7}{4} + (\frac{7}{4})^2 + \cdots + (\frac{7}{4})^{k-1}\}$$

$$= 7^k + \frac{18}{4} n^2 \cdot \frac{(\frac{7}{4})^k - 1}{(\frac{7}{4}) - 1}$$

$$\leq 7^k + 6n^2 (\frac{7}{4})^k$$

$$= 7^k + 6 \cdot 7^k$$

$$= 7 \cdot 7^k \qquad 7^k = n^{\log_2 7}$$

$$= 7 \cdot n^{\log_2 7}$$

$$= 7 \cdot n^{2.8} \qquad n^{2.49\cdots}$$

組合せアルゴリズムの評価

<u>計算時間</u>　　　　　　記憶容量

n：　グラフの規模　（点数）

T(n)：　点数nのグラフに対してかかる時間

又は

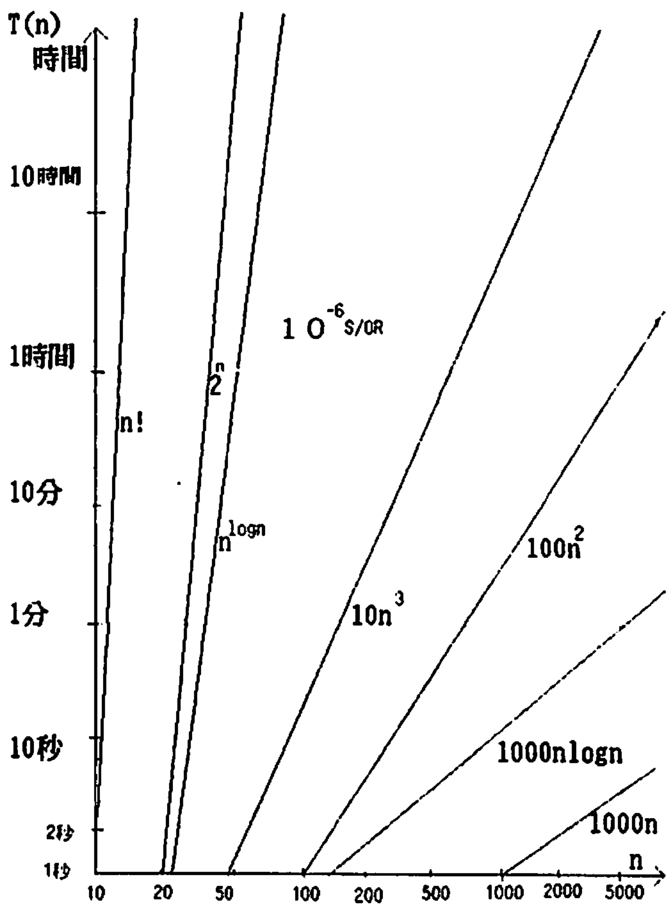
基本演算の総延数

(n) = O(f(n))：　高々f(n)のオーダ，　　　$g(n) \le cf(n)$

Ω(f(n))：　少なくともf(n)のオーダ，　$g(n) \ge cf(n)$

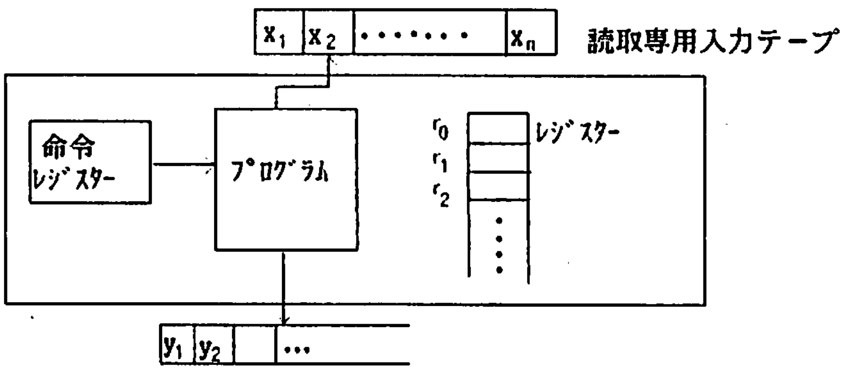
θ(f(n))：ぴったりf(n)のオーダ　，　$c_1f(n) \le g(n) \le c_2f(n)$

計算時間の上限			O				
下限			Ω				
正確な値			θ				
期待値	10	20	50	100	200	500	1000



計算のモデル

RAM(Random Access Machine)



プログラム

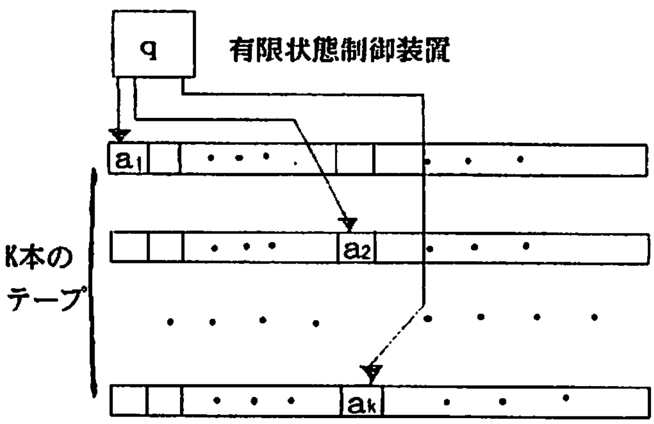
$$\left.\begin{array}{ll} \text{READ } X_i, & \text{WRITE } X_i \\ X_i := X_j + X_k & X_i := X_j - X_k \\ X_i := X[X_j] & X[X_j] := X_i \end{array}\right\} \text{からなる有限列}$$

IF $X_i = 0$ GOTO 1

命令

LOAD	JUMP
STORE	JGTZ
ADD	JZERO
SUB	
MULT	HALT
DIV	
READ	
WRITE	

テューリング機械



7個組（Q，T，I，δ，b，q_0，q_f）

1．Q　：状態の有限集合

2．T　：テープ記号の集合

3．I　：入力記号の集合で　$I \subset T$

4．b　：空白記号　$b \in T - I$

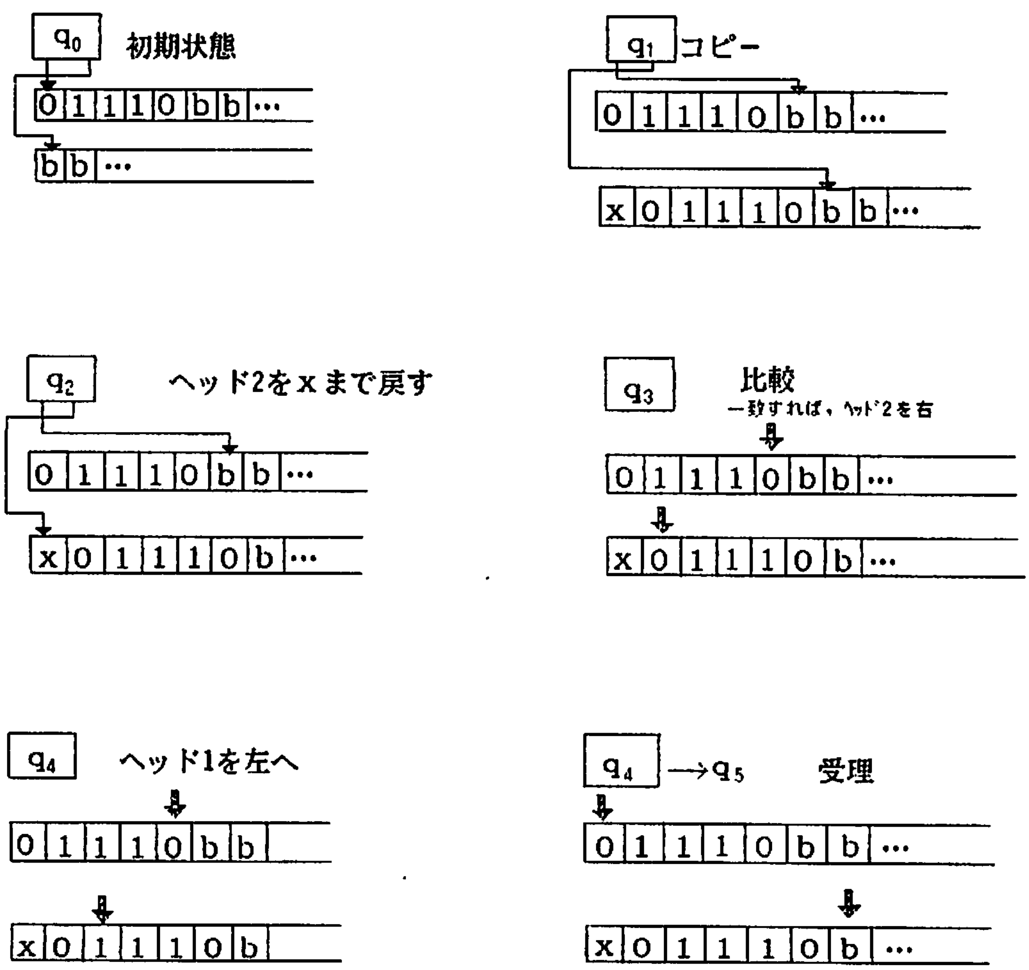
5．q_0：初期状態

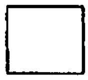
6．q_f：最終(受理)状態

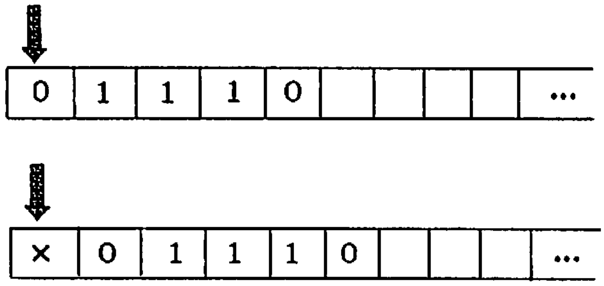
7．δ　：動作関数　　$Q \times T^k$の部分集合 $\rightarrow Q \times (T \times \{L, R, S\})^k$

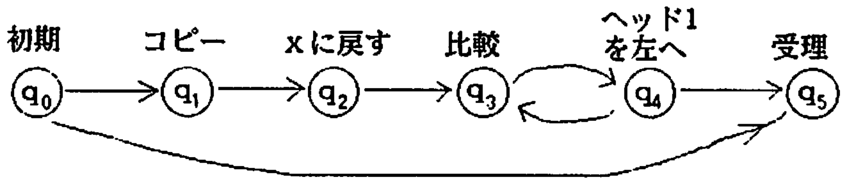
$$\delta(q, a_1, a_2, \cdots, a_k) = (q', a_1', d_1'), (a_2', d_2), \cdots, (a_k', d_k)$$

例　回文　01110　の判定









現在の状態	T^2		新記号とテープヘッド移動		新状態	備考
	テープ1	テープ2				
q_0	0	b	0,S	X,R	q_1	空なら受理 空でないなら，端記号Xを書いて，コピー状態q_1へ
	1	b	1,S	X,R	q_1	
	b	b	b,S	b,S	q_5	
q_1	0	b	0,R	0,R	q_1	テープ1をテープ2へコピー
	1	b	1,R	1,R	q_1	
	b	b	b,S	b,L	q_2	
q_2	b	0	b,S	0,L	q_2	テープ2のヘッドをXまで戻す
	b	1	b,S	1,L	q_2	
	b	×	b,L	X,R	q_3	
q_3	0	0	0,S	0,R	q_4	比較して一致すればヘッド2だけ右へ
	1	1	1,S	1,R	q_4	
q_4	0	0	0,L	0,S	q_3	ヘッド1を左へ右端まできていれば受理
	0	1	0,L	1,S	q_3	
	1	0	1,L	0,S	q_3	
	1	1	1,L	1,S	q_3	
	0	b	0,S	b,S	q_5	
	1	b	1,S	b,S	q_5	
q_5						受理状態

定理

（i） 問題Pがテューリング機械で$T(n)$時間で計算できる。

$\Longrightarrow$

PはRAMで$O(T(n))$時間で計算できる。

（ii） PがRAMで$T(n)$時間で計算できる。

$\Longrightarrow$

Pはテューリング機械で$O(T(n)^3)$ 時間で計算できる。

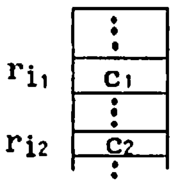
（iii） PがRAMで$O(S(n)$ 領域で計算

$\Longleftrightarrow$

Pがテューリング機械で$O(S(n)$ 領域で計算

#	#	i_1	#	c_1	#	#	i_2	#	c_2	#	#	…

RAM の TM表現



RAM のレジスター

アルゴリズムが無い問題
[計算機プログラムの停止問題]

P：プログラム

W：Pへの入力

$$R(P,W) = \begin{cases} 真：停止するとき \\ 偽：停止しないとき \end{cases}$$

停止判定アルゴリズムDがあると仮定

R (P,W)＝真のときYESを出力し,停止する.

〃　偽　NO　〃

P_0：Dのプログラムリスト

プログラム P_E

1) 入力 P のコピーを作る

2) P_D [P,P]

3) if P_D が YES を出力　then　ℓ : go to ℓ

R (P,P)が真のとき,プログラム P_E は入力 P に対して停止しない

R (P,P)が偽のとき,プログラム P_E は入力 P に対して停止する

Hilbert の第１０問題

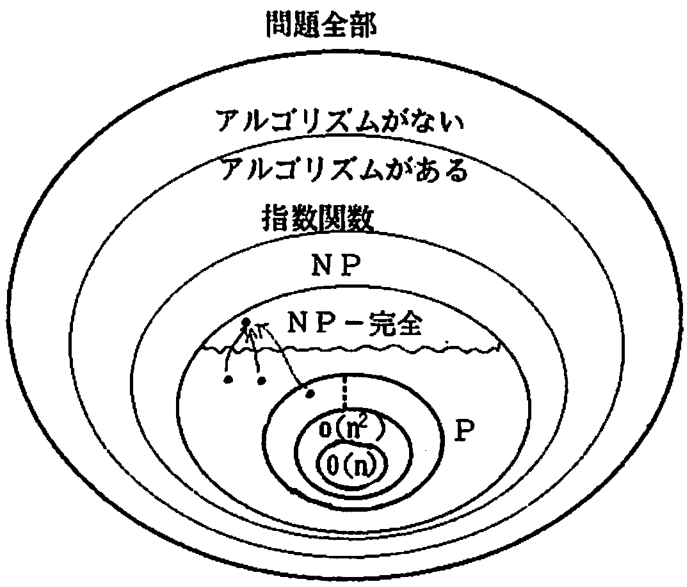
不定方程式 $P(x_1\cdots,x_n)=0$ が非負整数解を持つかどうか？

例

$$2x_1^2x_3^2 + 5x_2x_3 - x_2^3 - 12 = 0$$

解

$$\begin{cases} x_1 = 2 \\ x_2 = 1 \\ x_3 = 1 \end{cases}$$



ＮＰ－完全

非決定性テューリング機械（ＮＤＴＭ）

（Ｑ，Ｔ，Ｉ，δ，ｂ，q_0，q_f）

$$\delta : Q \times T^k \rightarrow 2^{Q \times (T \times \{L, R, S\})^k}$$

ＮＤＴＭの時間計算量　Ｔ(n)

長さｎのすべての受理される入力に対して，高々Ｔ(n)回で受理状態へ行く動作系列が存在

非決定性ＲＡＭ

choice （Ｓ）：集合Ｓから任意の１つを選ぶ

failure ：途中の choice での選択が失敗し，停止

success ：計算が成功

和積形論理式の充足可能性問題（ＳＡＴ）

$$F = (x \lor \bar{y} \lor z) \land (\bar{x} \lor \bar{z}) \land (y \lor \bar{z})$$

$x=1,\ y=1,\ z=1$ のとき $F=0$

$x=0,\ y=1,\ z=1$ のとき $F=1$　　　ＹＥＳ

$$F = (x \lor y) \land \bar{x} \land \bar{y}$$　　　ＮＯ

入力　　F　（変数を $x_1,\ x_2, \cdots x_m$ とする）

手続　　for　i:=1　to　m　do

　　　　　x_i :=choice (0,1) ;

　　　　if　F ($x_1, \cdots x_m$) =1　　then　　success

　　　　eise　　failure

$$T(n) = O(n)$$

$\hat{P}$　：（決定性）アルゴリズムで多項式時間で解ける問題

ＮＰ：　非決定性アルゴリズムで多項式時間で解ける問題

　　$P \subseteq NP$

　　「$P = NP$ か又は $P \neq NP$」未解決

問題 Q_1 から Q_2 へ多項式時間変換可能

　（$Q_1 \propto Q_2$）

（ｉ）Q_1 の任意の問題例Iを Q_2 の問題例 f(I)に変換する多項式時間決定性アルゴリズムが存在

（ii）Q_1 におけるＩに対する答えが ＹＥＳ

$$\Leftrightarrow$$

Q_2 における f(I)に対する答えが　ＹＥＳ

Qが ＮＰ困難 ： $^{\forall}Q_J \in$ ＮＰ $Q_J \propto Q$

Qが ＮＰ完全 ： Q ∈ ＮＰ かつ QはＮＰ困難

Cookの定理 ： ＳＡＴ は ＮＰ完全

（略称） ＳＡＴ ∈ ＮＰ

次のような多項式時間決定性アルゴリズムを作る

入力 ： $^{\forall}Q \in$ ＮＰ， Qの任意の問題例Ｉ

出力 ：和積形論理式 Ｅ（Q，Ｉ）

但し

Ｅ（Q，Ｉ）は充足可能

⇔

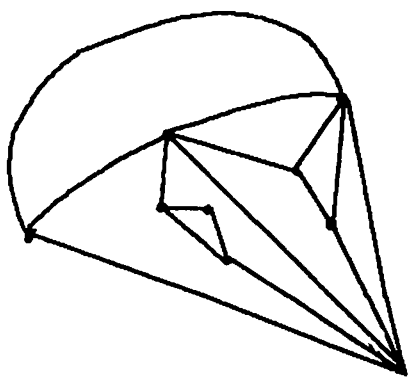
Q におけるＩの答えが ＹＥＳ

ＮＰ完全性の証明例

クリーク問題

入力 ：グラフＧ＝（Ｖ，Ｅ），と 整数k

出力 ：K_kがあれば ＹＥＳ，なければ ＮＯ

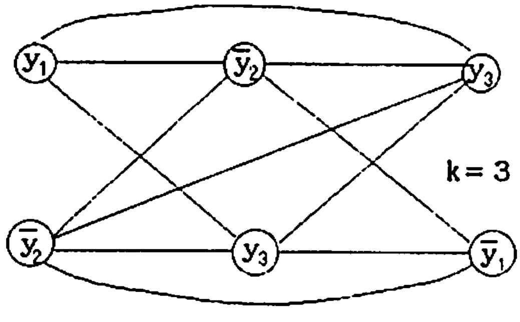


k= 4

ステップ１： クリーク問題 ∈ ＮＰ

ステップ２： ＳＡＴ ∝ クリーク

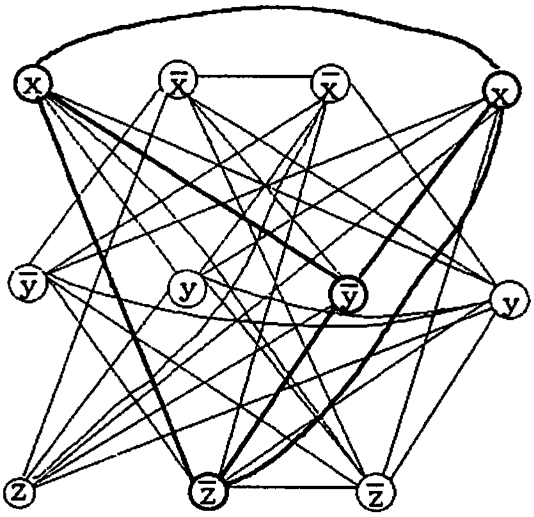
$F=(y_1 \vee y_2) \wedge (y_2 \vee y_3) \wedge (y_3 \vee y_1)$



k= 3

F が充足可能 ⇔ G に K_3 がある

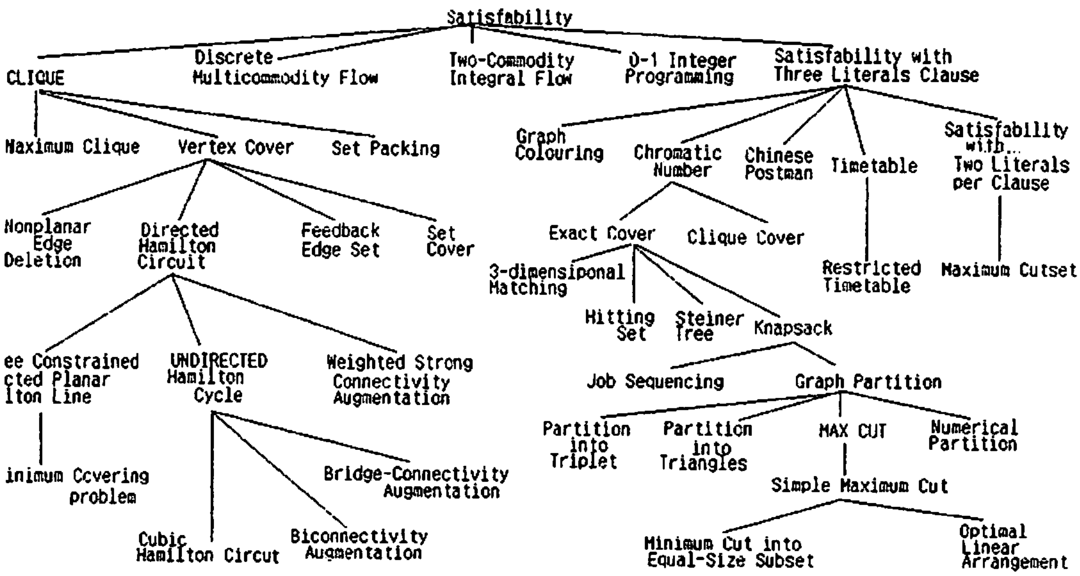
$$F=(x\vee\bar{y}\vee z)\wedge(\bar{x}\vee y\vee\bar{z})$$
$$\wedge(\bar{x}\vee\bar{y}\vee\bar{z})\wedge(x\vee y)$$



$x=1$

$y=0$

$z=0$



ＮＰ－完全問題

FIG　NP-Complete　ploblems　菊池　卒論

複雑さによる分類

第１グループ：多項式時間算法が見つけられているもの

平面グラフの判定	$O(n)$
連結成分への分解	$O(m)$
強連結成分への分解	$O(m)$
最小木	$O(MIN(n^2, m\log\log n))$ $O(m \log B(m,n))$
最短路	$O(n^2)$　　$O(m+n\log)$
最大マッチング	$O(n^{2.5})$　　$mn\log n\{t(m,n)+n\log n\}$
最大フロー	$O(n^3)$
オイラーグラフの判定	$O(m)$

第２グループ：指数関数時間がかかることが判明いている

全ての木の発生

全ての道の発生　　　$d=\max\{m/n\}$

$t(m,n)=o(m \log \log \log_d n)$

$3(m,n)=\min\{i \mid \log^{(i)} n \leq m/n\}$

第３グループ　：ＮＰ－完全

大部分の組合せ問題

充足可能性問題	整数計画問題
独立点集合	辺彩色
ハミルトン閉路	最小種数
彩色数	ネットワーク信頼性
帰還点集合	グラフ分割
・	ナップザック問題
・	
・　無数　何百～何千	

第４グループ：　１，２，３のどれに入るか現在のところ不明

グラフ同形問題

部分グラフ位相同形問題

ＬＰ　$O(((m+n)n^2+(m+n)^{1.5n})L)$

第５グループ：アルゴリズムがないもの

ＤＦＴ：離散的フーリェ変換

ＭＭ　：行列積　　　　　　　ＭＮＦ：最大フロー

ＰＭ　：パターンマッチング　　ＧＭ　：マッチング

ＳＣ　：強連結成分

ＰＴ　：平面性判定

ROBERT ENDRE TARJAN

n：点数

m：枝数

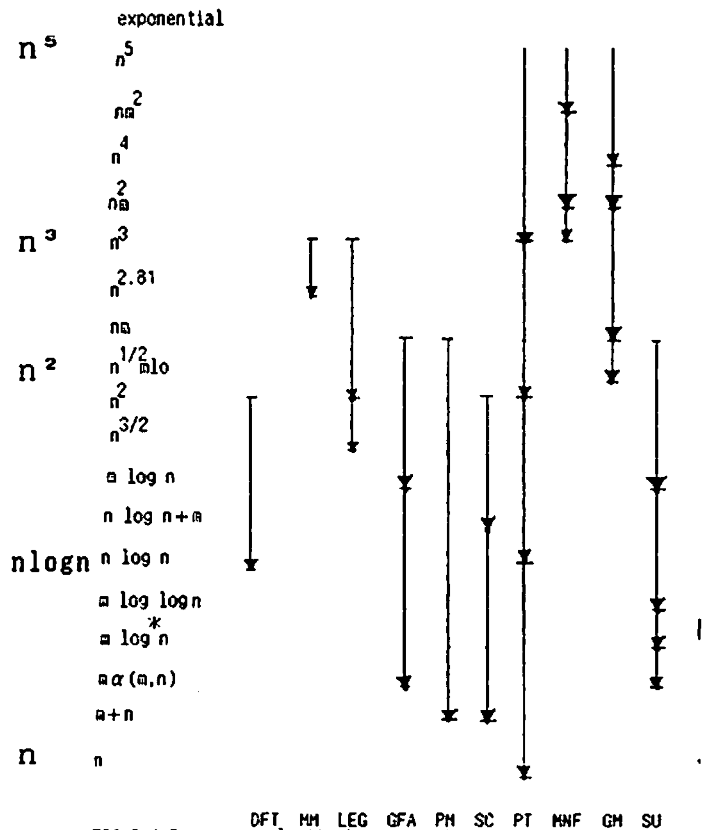


FIG.5.1.Recent complexity improvements.
n=size(number of vertices in graph problems).
m=second parameter (number of edges in graph problems).

アルゴリズム改良の歴史

Cooley & Tukey　1965

$T(n) = 2T(n/2) + n$

nヶ

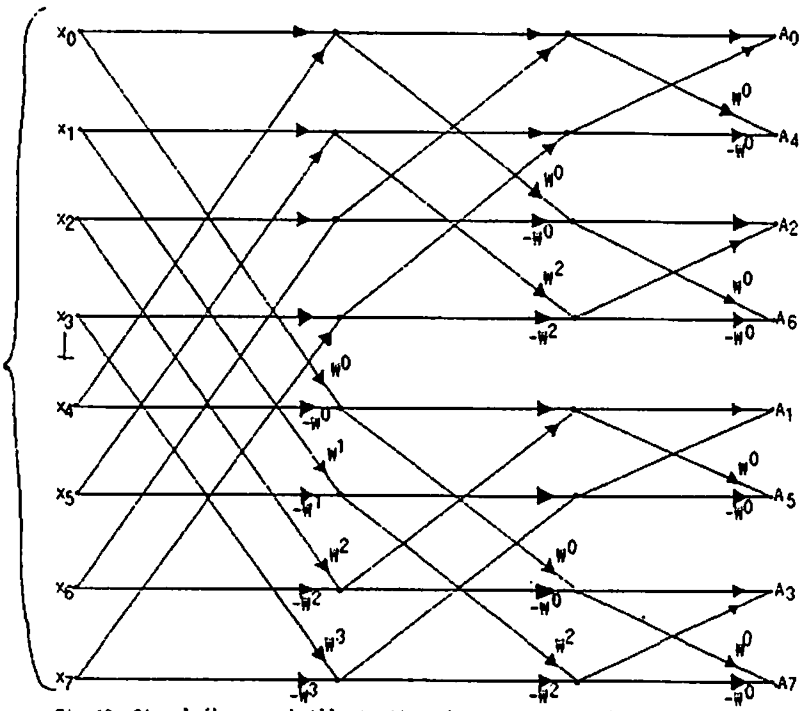


Fig.10. Signal flow graph illustrating the computation of the DFT when the operations involved are completely reduced to multiplications and additions.

FFT

$$b_k = \sum_{i=0}^{n-1} a_i \, w^{ik}$$

w :

$O(n2) \rightarrow O(n \log n)$

アルゴリズム改良のトリック

1．データ構造の工夫

　　リスト　　ヒープ

　　２－３木　　MENBER, INSERT　　O(nlogn)

　　　　　　　　DELETE, UNION

　　　　　　　　FIND, MIN

2．最帰的呼び出し　(Recursion)

　　Dynamic programming　　(例．　Ford-Bellman法)
　　　(動的計画法)

　　Divide - and - conquer　　(例．　marge sart)
　　　(分割統治法)　　　　　　　例　　行列積
　　　　　　　　　　　　　　　　例　　FFT

3．最適化手法

　　Greedy method　　(最小木)

　　Augmentation　　(例　最大フロー，最大マッチング)

4．探索 (searching)

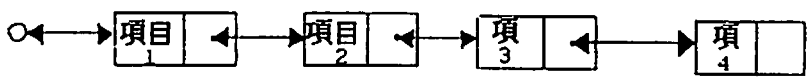
　　Depth-first　　(平面性判定)

　　Breadth-first　　(Dijkstra法)

5．分割

データ構造

リスト



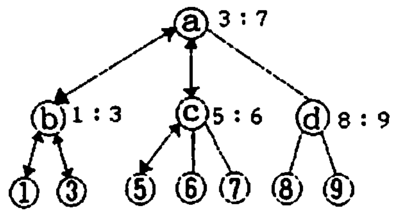
	NAME	NEXT	前
0	－	1	
1	項目１	3	0
2	項目４	0	4
3	項目２	4	1
4	項目３	2	3

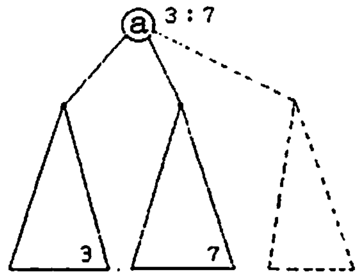
両方向リスト

	START
倉庫１	0
倉庫２	5
	⋮

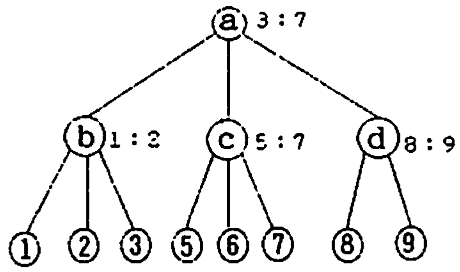
MENBER(x)	|List|
INSERT(x)	O(1)
DELETE(x)	O(n)
UNION	O(1)
MIN	O(List)

2-3木　：子供が2,3人で、rootからleafまでの長さが同じ



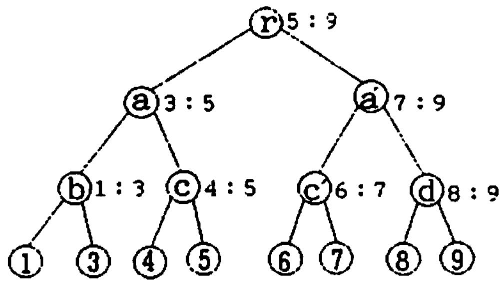


MENBER (2)
INSERT (2)

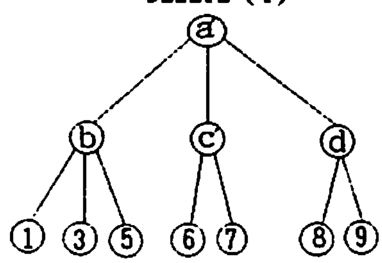


$\boxed{\log\ n}$

INSERT (4)



DELETE (4)



最大値最小値を見つけるアルゴリズム

S：集合，　｜S｜＝ n

◎ MAXを見つける単純な方法

begin

MAX ← Sのひとつの要素；

```
for  S中の他のすべての要素 x  do
    if  x > MAX  then  MAX ← x
end
```

比較　$(n-1) + (n-2) = 2n-3$ 回

（$n-1$ ← MAX，$n-2$ ← MIN）

◎　分割統治法による方法

```
procedure    MAXMIN (S)
if |S|=2  then
   begin
     S = {a,b} とする；
     return (MAX(a,b), MIN(a,b))
   end
else
   begin
     SをS₁とS₂に分割；
     (max1, min1) ← MAXMIN(S₁)；
     (max2, min2) ← MAXMIN(S₂)；
     return (MAX(max1, max2), MIN(min1, min2))
   end
```

$$T(n) = \begin{cases} 1 & : n = 2 \\ 2T\left(\frac{n}{2}\right) + 2 & : n > 2 \end{cases}$$

$$T(n) = \frac{3}{2}n - 2$$

（証）　帰納法　（nは2の巾とする）

ベース　$T(2) = \frac{3}{2} \cdot 2 - 2 = 1$

$n = m$で成立を仮定

$$
\begin{aligned}
T(2m) &= 2T(m) + 2 \\
&= 2\left(\frac{3}{2}m - 2\right) + 2 \\
&= \frac{3}{2}(2m) - 2 \\
&\le \frac{3}{2}(2m) + 2
\end{aligned}
$$

Q. E. D

再帰, Divide-and-conqure, バランス

ソート

入力 : $x_1, x_2, \cdots, x_n$ 　　　5 7 2 8 6 1 4 3

出力 : $y_1 \le y_2 \le \cdots \le y_n$ 　　　1 2 3 4 5 6 7 8

MERGE (S, T)

$$
\left.\begin{array}{l} S = 2\ 5\ 7\ 8 \\ T = 1\ 3\ 4\ 6 \end{array}\right\} \Rightarrow 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8
$$

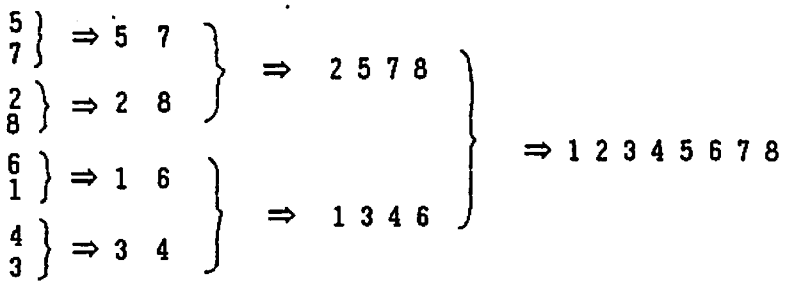
高々 $|S| + |T| - 1$ 回の比較

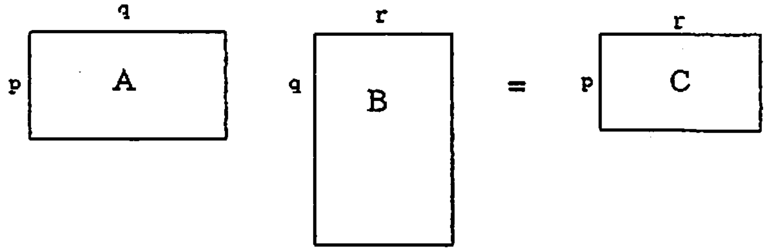
マージソート

```
procedure  SORT (i,j)
if  i=j  then  return  x_i
else
  begin
    m ← ⌊(i + j - 1)⌋ / 2  ;
    S ← SORT (i, m)  ;
    T ← SORT (m+1, j) ;
    return  MERGE (S, T)
  end
```

$$
T(n) = \begin{cases} 0 & : n = 1 \text{のとき} \\ 2T\left(\frac{n}{2}\right) + n - 1 & : n > 1 \text{のとき} \end{cases}
$$

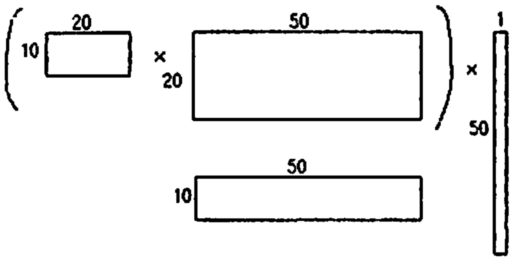
$$
\begin{aligned}
T(n) &= 2\left\{2T\left(\frac{n}{2^2}\right) + \frac{n}{2} - 1\right\} + n - 1 \\
&= 2^2\, T\left(\frac{n}{2^2}\right) + n - 2 + n - 1 \\
&\cdots\cdots\cdots\cdots\cdots\cdots \\
&\le 2^{\log n}\, T\left(\frac{n}{2^{\log n}}\right) \\
&\quad + \underbrace{n + n + \cdots\cdots + n}_{\log_2 n \text{個}} \\
&= n \log n
\end{aligned}
$$





$c_{ij} = \sum_{k=1}^{q} a_{ik}\, b_{kj}$ × q回 + q-1回

C 合計 pqr回 + p(q-1)r ≒ 2 pqr



10×20×50= 10000
10×50× 1= 500
計 10500

(M_1) (M_2 M_3 M_4)	1 × 2
(M_1 M_2) (M_3 M_4)	1 × 1
(M_1 M_2 M_3) (M_4)	2 × 1
	計 5通り

(M_1) (M_2 M_3 M_4 M_5)	1 × 5 = 5
(M_1 M_2) (M_3 M_4 M_5)	1 × 2 = 2
(M_1 M_2 M_3) (M_4 M_5)	2 × 1 = 2
(M_1 M_2 M_3 M_4) (M_5)	5 × 1 = 5
	計 14通り

$X(n)$: nヶ の行列積のカッコの入れ方

$$\begin{cases} X(1) = 1 \\ X(n) = \sum_{i=1}^{n-1} X(i)\,X(n-i) & n > 1 \end{cases}$$

$$X(n+1) = \frac{1}{n+1}\binom{2n}{n}$$

catalan 数

$$X(n) \geq 2^{n-2}$$

動的計画法

小さな部分問題からより大きな部分問題を解の表を利用して解いていく

行列積の計算手間の最小化

$$M = (M_1)\Big(\underset{20\times 50}{M_2}\ \big(\underset{50\times 1}{M_3}\ \ \underset{1\times 100}{M_4}\big)\Big)$$

$$\begin{array}{rr} & 50\times 1\times 100 = 5000 \\ & 20\times 50\times 100 = 100000 \\ +) & 10\times 20\times 100 = 20000 \\ \hline & 125000 \end{array}$$

$$r_{i-1}\,\overbrace{\overset{r_i}{\boxed{M_i}}\ \ r_i\ \overset{r_{i+1}}{\boxed{M_{i+1}}}\ \cdots\ r_{k-1}\,\overset{r_k}{\boxed{M_k}}}^{m_{ik}} \times \overbrace{\boxed{M_{k+1}}\ \cdots\ \overset{r_j}{\boxed{M_j}}}^{m_{k+1,j}}$$

```
begin
  for  i ← 1  until  n  do   mii ← 0  ;
  for  ℓ ← 1  until  n − 1  do
  for  i ← 1  until  n − ℓ  do
    begin
      j ← i + ℓ ;
      mij ← MIN  (mik + mk+1,j + ri-1 rk rj)
          i≤k<j
    end
  write   m1n
end
```

$$O(n^3) \qquad \Big(\underbrace{\overbrace{(M_i\ \ M_2 \cdots M_k)}^{m_{ik}}}_{r_{i-1}\times r_k}\ \underbrace{\overbrace{(M_{k+1}\ \cdots\ M_j)}^{m_{k+1,j}}}_{r_k\times r_j}\Big)$$

$$M = \left(\left({}_{10}\overset{20}{M_1}\right){}_{20}\overset{50}{M_2}\ {}_{50}\overset{1}{M_3}\right)\left({}_{1}\overset{100}{M_4}\right)$$

$m_{11}=0$	$m_{22}=0$	$m_{33}=0$	$m_{44}=0$
$m_{12}=10000$	$m_{23}=1000$	$m_{34}=5000$	
$m_{13}=1200$ $k=1$	$m_{24}=3000$ $k=3$		
$m_{14}=2200$ $k=3$			

$$m_{13} = \mathrm{MIN}\left\{\begin{array}{l} \overset{0}{m_{\underline{11}}} + \overset{1000}{m_{23}} + 10\times20\times1 = 1200 \\ m_{12} + m_{33} + 10\times50\times1 = 10500 \end{array}\right.$$

$$m_{14} = \mathrm{MIN}\quad \begin{array}{l} \overset{0}{m_{11}} + \overset{3000}{m_{24}} + 10\times20\times100 = 23000 \\ \overset{10000}{m_{12}} + \overset{5000}{m_{34}} + 10\times50\times100 = 65000 \\ \overset{1200}{m_{\underline{\underline{13}}}} + \overset{0}{m_{44}} + 10\times\ 1\times100 = 2200 \end{array}$$

(例)

$$M = \left(\left({}_{1}\overset{5}{M_1}\quad {}_{5}\overset{4}{M_2}\right){}_{4}\overset{3}{M_3}\right)\left({}_{3}\overset{2}{M_4}\right)$$

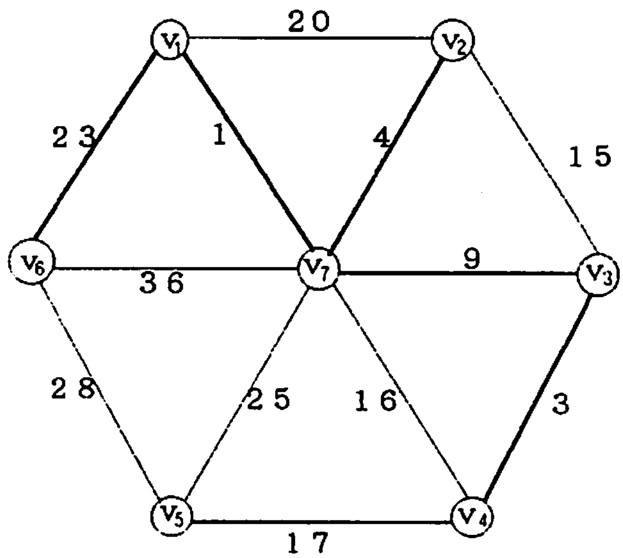
$m_{11}=0$	$m_{22}=0$	$m_{33}=0$	$m_{44}=0$	
$m_{12}=20$	$m_{23}=60$	$m_{34}=24$		
$m_{13}=32$ $k=2$	$m_{24}=64$ $k=2$			
$m_{14}=38$ $k=3$				

$$m_{13} = \mathrm{MIN}\begin{cases} \overset{0}{m_{11}} + \overset{60}{m_{23}} + 1\times5\times3 = 75 \\ \underset{20}{m_{12}} + \underset{0}{m_{33}} + 1\times4\times3 = 32 \end{cases}$$

$$m_{24} = \mathrm{MIN}\begin{cases} \overset{0}{m_{22}} + \overset{24}{m_{34}} + 5\times4\times2 = 64 \\ \underset{60}{m_{23}} + \underset{0}{m_{44}} + 5\times\underset{30}{3\times2} = 90 \end{cases}$$

$$m_{14} = \mathrm{MIN}\begin{cases} \overset{0}{m_{11}} + \overset{64}{m_{24}} + 1\times5\times2 = 74 \\ \overset{20}{m_{12}} + \overset{24}{m_{34}} + 1\times4\times2 = 52 \\ \overset{32}{m_{13}} + \overset{0}{m_{44}} + 1\times3\times2 = 38 \end{cases}$$

最小木



1＋3＋4＋9
＋17＋23＝57

キューQ

枝	コスト
(v_1, v_7)	1
(v_3, v_4)	3
(v_2, v_7)	4
(v_3, v_7)	9
(v_2, v_3)	15
(v_4, v_7)	16
(v_4, v_5)	17
(v_1, v_2)	20
(v_1, v_6)	23
(v_5, v_7)	25
(v_5, v_6)	28
(v_6, v_7)	36

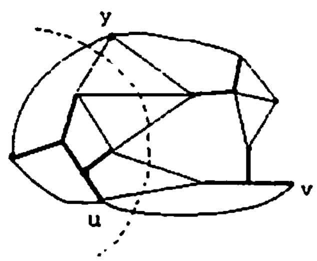
```
begin
  T ← φ ;
  VS ← φ ;
  キューQを作る；
  for  各点  v∈V  do  { v }をVSに加える；
  while  |VS|＞1  do
     begin
        コスト最小のQの枝 ( v,w )を選ぶ；
        Qから ( v,w )を除去；
        if  W₁∋v と W₂∋wが異なる then
        begin
          VS 中の W₁とW₂ を W₁∪W₂ でおきかえる；
          枝 ( v,w )を集合Tに加える
        end
     end
end
```

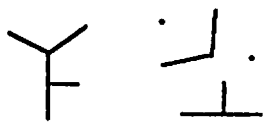
	W	NEXT
v_1	W_1	2
v_2	W_1	7
v_3	W_3	4
v_4	W_3	0
v_5	W_5	0
v_6	W_6	0
v_7	W_1	0

	開始	大きさ
W_1	1	3
W_2		
W_3	3	2
W_4		
W_5	5	1
W_6	6	1
W_7		
	4	

ソート	O（E log V）
集合	O（V log V）
	O（E log V）　　O（E loglog V）

補題





を含む最小木があるとき、

………………のコスト最小の枝を含む最小木がある

（証）

$c(x, y) \leq c(u, v)$

青 ＋（x，y）を含む最小木は存在しないが、青 ＋（u，v）を含む最小木Tが存在したとする。　　$T' = T + (x, y) - (u, v)$

$c(T') \leq c(T)$

$W_1 \longrightarrow \boxed{V_1|W_1} \longrightarrow \boxed{V_2|W_1} \longrightarrow \boxed{V_7|W_1} \longrightarrow \boxed{O}$

$W_3 \longrightarrow \boxed{V_3|W_3} \longrightarrow \boxed{V_4|W_3} \longrightarrow \boxed{O}$

$W_5 \longrightarrow \boxed{V_5|W_5} \longrightarrow \boxed{O}$

$W_6 \longrightarrow \boxed{V_6|W_6} \longrightarrow \boxed{O}$

$W_1 \cup W_3$ にかかる時間

$\mathrm{MIN}\{|W_1|, |W_3|\}$

どの箱も高々 $\log_2 |v|$ 回しか書き換えられない

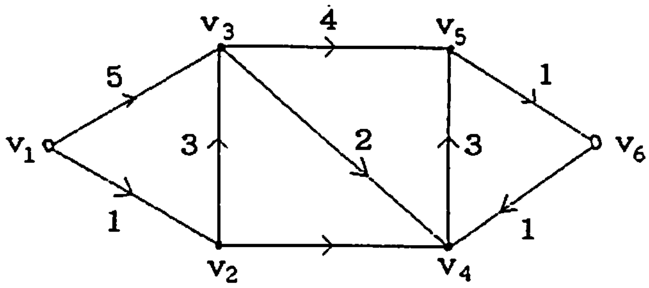
∵もしも $(\log_2 v)+1$回以上書き換えられた箱があったとすると、その箱を含むリスト（即ち集合）の大きさは $2^{(\log 2v)+1}$ 以上になるはず．

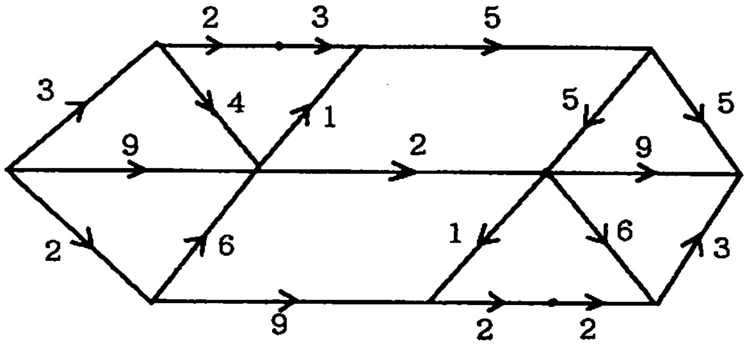
しかし $2^{(\log v)+1} > V$ なので矛盾

・箱はV個あるので、集合の管理はV log V時間でできる

最短路を求めるDijkstra法

```
begin
  P ← {v₁} ;
  l (v₁) ← O ;
  for 各v∈V − {v₁}   do l(v) ←d(v₁, v) ;
  while P≠V  do
    begin
      V − Pの中の点wで l(w) が最小の点を選ぶ ;
      P←P∪ {w} ;
      for 各v∈V − P  do
        l (v) ←MIN (l(v) , l(w) +d (w, v) )
    end
end
```



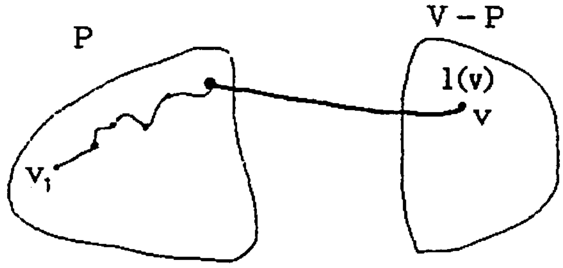


P：　v_1からの距離が確定している点の集合

V－P：　v_1からの距離が確定していない点の集合

$\forall_v \in P$　　　l(v)：　$v_1 \rightarrow v$の最短路の長さ

$\forall_v \in V-P$　　l(v)：　v_1からPの点だけを通って（V－Pの点は一つも通らずに）　vへ行く最短路の長さ



補題

$w \in V - P$ かつ $l(w)$ が最小

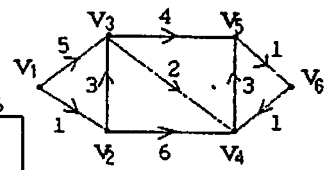
$\Rightarrow$

$l(w)$ は既に $v_1 \rightarrow w$ の真の最短距離

(証) $l(w) > d(v_1, w) = l(x) + \underset{\substack{\text{\rotatebox{90}{$\leq$}} \\ 0}}{d(x, w)}$

矛盾

グラフの表現



隣接行列

	v_1	v_2	v_3	v_4	v_5	v_6
v_1	0	1	5			
v_2		0	3	6		
v_3			0	2	4	
v_4				0	3	
v_5					0	1
v_6				1		0

距離行列

隣接リスト

$v_1 \longrightarrow$ [v_2, 1 | •] $\longrightarrow$ [v_3, 5 | •] ⏚

$v_2 \longrightarrow$ [v_4, 6 | •] $\longrightarrow$ [v_3, 3 | •] ⏚

$v_3 \longrightarrow$ [v_4 2 | •] $\longrightarrow$ [v_5, 4 | •] ⏚

$v_4 \longrightarrow$ [v_5, 3 | •] ⏚

$v_5 \longrightarrow$ [v_6, 1 | •] ⏚

$v_6 \longrightarrow$ [v_4, 1 | •] ⏚

O (| E |)
メモリー

Dijkstra法の計算時間

while ループ1回当たり

	MIN	O(V)
+)	ラベルの書換	O(V)
	小計	O(V)

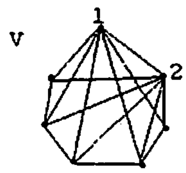
while　ループは　V回まわる

全体で　　$O(V^2)$ 時間

$E = V^{1+\varepsilon}$

$0 < \varepsilon < 1$

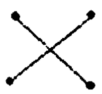
完全グラフ



$$E = \frac{V(V-1)}{2}$$

$E = CV$

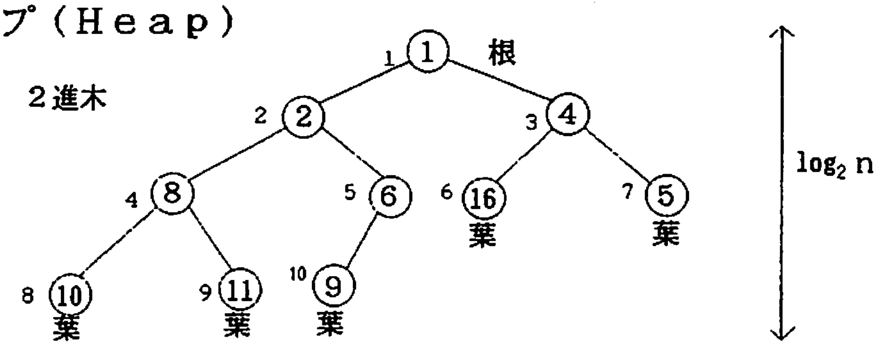
sparseなグラフ



$$|E| = \binom{|V|}{2} = \frac{V(V-1)}{2}$$

ヒープ (Heap)

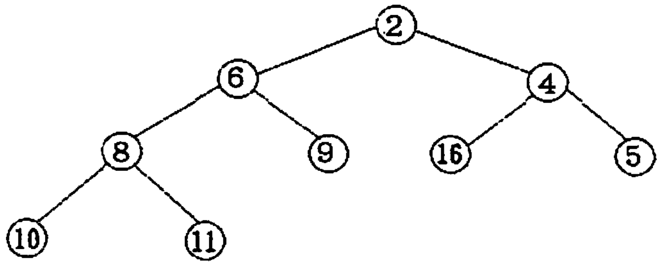
2進木



	1	2	3	4	5	6	7	8	9	10
A	1	2	4	8	6	16	5	10	11	9

A[i] の子供は A[2i] と A[2i+1]

A[i] の親　は　　A $[\lfloor \frac{i}{2} \rfloor]$



1	2	3	4	5	6	7	8	9	10
2		4	8	6	16	5	10	11	9

2	6	4	8		16	5	10	11	9

2	6	4	8	9	16	5	10	11	

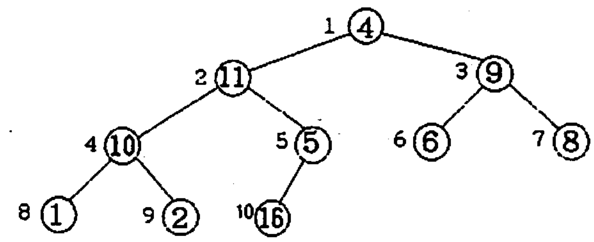
i の子供は　2i と 2i+1

i の親は　$\lfloor \frac{i}{2} \rfloor$

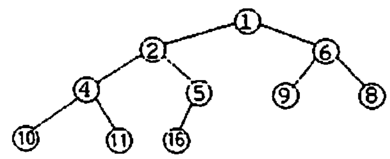
	value	LS	RS	Fath
1	1	2	3	
2	2	4	5	1
3	4	6	7	1
4	8	8	9	2
5	6	10		2
6	16			3
7	5			3
8	10			
9	11			
10	9			

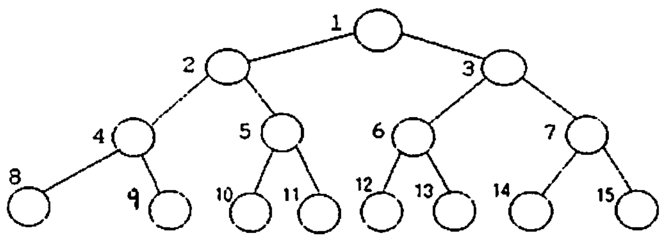
```
procedure  M I N :
   begin
     M I N  ←  A [ 1 ] ;
     i  ←  1 ;
     while   i が葉でない    do
       begin
         i の子供のうち小さい方を k とする ;
         A [ i ] ←  A [ k ] ;
         i  ←  k
       end
   end
```

```
procedure  MODIFY (i)
  while  i≠1 かつ A[i] < A[⌊i/2⌋]  do
    begin
      A[i] ⟷ A[⌊i/2⌋] ;
      i ⟵ ⌊i/2⌋
    end
```



1	2	3	4	5	6	7	8	9	10
				5	6	8	1	2	16
			10	5	6	8	1	2	16
			1	5	6	8	10	2	16
		9	1	5	6	8	10	2	16
	11	6	1	5	9	8	10	2	16
	1	6	11	5	9	8	10	2	16
4	1	6	2	5	9	8	10	11	16
1	4	6	2	5	9	8	10	11	16
1	2	6	4	5	9	8	10	11	16





```
Procedure    HEAPIFY (i,j) :
 if   iは葉でなく, iより小さな子がある  then
   begin
    kをその小さいほうの子とする;
    A[i] ←——→ A[k] ;
    HEAPIFY (k,j)
   end

procedure   BUILDHEAP :
    for  i ← n  step -1  until 1  do HEAPIFY (i,n)
```

Dijkstra法の計算時間

MIN　　　　　　$O(|V|)$
ラベルの書き換え　$O(|V|)$

$\}$ $|V|$回繰り返し
計　$O(|V|^2)$

ヒープを使えば

		小計
ヒープの構成		$O(V)$
MIN	$O(\log\|V\|)$ …	$O(V \log V)$
ラベルの書き換え	$O(d(v))$ …	$O(E)$
〃 によるヒープの変更	$O(\log V)$	$O(E \log V)$
	合計	$O(E \log V)$
	Tarjan 他 1984年	$O(E + V \log V)$

ヒープの構成

高さ i の点にHEAPIFYをした時　$O(i)$

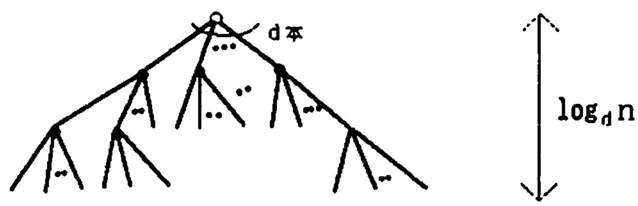
高さ i の点は 高々 $\left\lceil \frac{n}{2^{i+1}} \right\rceil$ 個

合計

$$\sum_{i=1}^{h} i \frac{n}{2^i} = O(n)$$

$$\left(\because \sum_{i=1}^{h} i r^i = \frac{-h r^{h+1}}{1-r} + \frac{r(1-r^n)}{(1-r)^2} \right)$$

d-ヒープ



$d = \frac{E}{V}$

$dV\log_d V = E \frac{\log v}{\log d} = E \frac{\log V}{\log E - \log V} = O(E)$

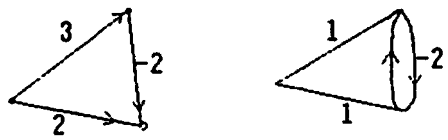
$E = V^{1+\alpha}$　　$\alpha > 0$ のとき

$E\log_d V =$　　　〃

$E = V\log V$ のとき $E = V^{1+\alpha}$, $\alpha > 0$ なる α は ない

$E = CV$

負枝があるとき　Dijkstra法は動くか?



全点間の最短路を求めるにはどうする?

Dijkstra法を各点について実行?

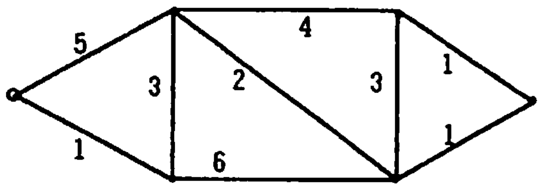
V・E log V　　時間

完全グラフなら　$V^3 \log V$

⇩

V^3

無向グラフのとき、どうしたらよいか?



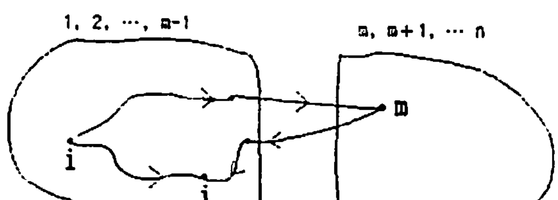
全点間の最短路（Floyd － Warshall法）

$U_{ij}^{(m)}$ ＝点m，m＋1，…，nを通らずにiからjに行く最短路長

$U_{ij}^{(1)} = a_{ij}$

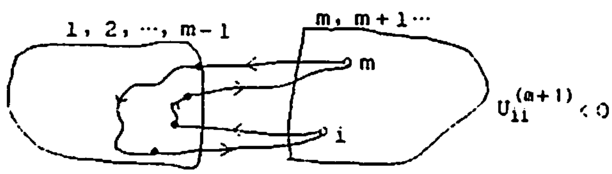
$U_{ij}^{(m+1)} = \mathrm{MIN}\{\, U_{ij}^{(m)},\ U_{im}^{(m)} + U_{mj}^{(m)} \,\}$

$U_{ij}^{(n+1)} = U_{ij}$



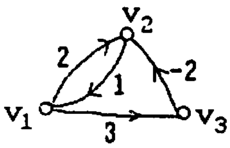
計算時間　$O(n^3)$

閉路存在判定



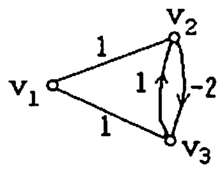
i：負閉路で最大の点　　　$m+1 \leq i \leq n$

m：負閉路で最大の次の点



$$U^{(1)}= \begin{array}{cc} & \begin{array}{ccc}1&2&3\end{array} \\ \begin{array}{c}1\\2\\3\end{array} & \begin{bmatrix}0&2&\underline{3}\\ \underline{1}&0&\infty\\ \infty&-2&0\end{bmatrix}\end{array} \qquad U^{(2)}= \begin{array}{cc} & \begin{array}{ccc}1&2&3\end{array} \\ \begin{array}{c}1\\2\\3\end{array} & \begin{bmatrix}0&2&3\\ \underline{1}&0&4\\ \infty&\underline{-2}&0\end{bmatrix}\end{array}$$

$$U^{(3)}= \begin{array}{cc} & \begin{array}{ccc}1&2&3\end{array} \\ \begin{array}{c}1\\2\\3\end{array} & \begin{bmatrix}0&2&\underline{3}\\ 1&0&4\\ -1&\underline{-2}&0\end{bmatrix}\end{array} \qquad U^{(4)}= \begin{array}{cc} & \begin{array}{ccc}1&2&3\end{array} \\ \begin{array}{c}1\\2\\3\end{array} & \begin{bmatrix}0&1&3\\ 1&0&4\\ -1&-2&0\end{bmatrix}\end{array}$$



$$U^{(1)} = U^{(2)} = \begin{array}{cc} & \begin{array}{ccc}1&2&3\end{array} \\ \begin{array}{c}1\\2\\3\end{array} & \begin{bmatrix}0&\underline{1}&1\\ \infty&0&\underline{-2}\\ \infty&\underline{1}&0\end{bmatrix}\end{array} \qquad U^{(3)} = \begin{array}{cc} & \begin{array}{ccc}1&2&3\end{array} \\ \begin{array}{c}1\\2\\3\end{array} & \begin{bmatrix}0&1&-1\\ \infty&0&\underline{-2}\\ \infty&\underline{1}&-1\end{bmatrix}\end{array}$$

$$U^{(4)} = \begin{array}{cc} & \begin{array}{ccc}1&2&3\end{array} \\ \begin{array}{c}1\\2\\3\end{array} & \begin{bmatrix}0&0&-1\\ \infty&-1&-2\\ \infty&1&-1\end{bmatrix}\end{array}$$

［定理］　負閉路がある

⇔

あるｉとｍに対し $u_{ii}^{(m)} < 0$

（$1 \le i \le n$, $1 \le m \le n$）

$f_{ij}^{(m)}$ ＝点m, m＋1, …nを通らずにｉからｊへ行く最短路の点で最大の点（≠ｉ，ｊ）

$$F^{(1)}= \begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix} \qquad F^{(2)} = \begin{bmatrix}0&0&0\\0&0&1\\0&0&0\end{bmatrix} \qquad P_{23} = P_{21}\,P_{13}$$

$$F^{(3)}= \begin{bmatrix}0&0&0\\0&0&1\\2&0&0\end{bmatrix} \qquad F^{(3)} = \begin{bmatrix}0&3&0\\0&0&1\\2&0&0\end{bmatrix}$$

（＊）

$$P_{ij} = \begin{cases} P_{i\,f_{ij}} \cdot P_{f_{ij}\,j} & : \ f_{ij} \neq 0 \ \text{のとき} \\ i \cdot j & : \ f_{ij} = 0 \ \text{のとき} \end{cases}$$

$$P_{12} = P_{13} P_{32}$$
$$= 1 \cdot 3 \cdot 2$$

再帰呼出しをスタックで実現

mに関する帰納法で(*)が証明できる

1.° $m=1$のとき, $F^{(1)} =$ で,明らかに成立

2.° m 以下で成立と仮定

3.° $u_{ij}^{(m+1)} = \mathrm{MIN}\,\{u_{ij}^{(m)},\ u_{im}^{(m)} + u_{mj}^{(m)}\}$

場合1: $u_{ij}^{(m+1)} = u_{im}^{(m)}$ のとき.

$f_{ij}^{(m+1)} = f_{ij}^{(m)},\ f^{(m+1)}(i, f_{ij}) = f^{(m)}(i, f_{ij}),\ f^{(m+1)}(f_{ij}, j) = f^{(m)}(f_{ij}, j)$

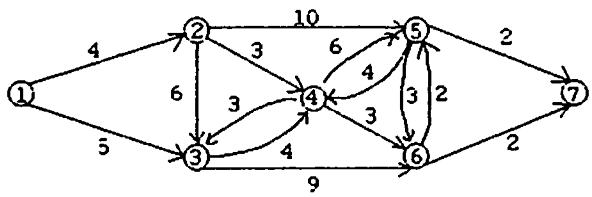
(*)が成立

場合2: $u_{ij}^{(m+1)} = u_{im}^{(m)} + u_{mj}^{(m)}$ のとき. $f_{ij}^{(m+1)} = m$

$f_{ij} = m$ $\qquad P_{ij} = P_{im}P_{mj}$

$f^{(m+1)}(i, m) = f^{(m)}(i, m),\ f^{(m+1)}(m, j) = f^{(m)}(m, j)$

$\because\ u_{im}^{(m+1)} = u_{im}^{(m)}$



$$U^{(1)} = U = \begin{array}{c} \\ 1\\2\\3\\4\\5\\6\\7 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{bmatrix} 0 & 4 & 5 & & & & \\ & 0 & 6 & 3 & 10 & & \\ & & 0 & 4 & \infty & 9 & \\ & & 3 & 0 & 6 & 3 & \\ & & & 4 & 0 & 3 & 2 \\ & & & & 2 & 0 & 2 \\ & & & & & & 0 \end{bmatrix} \end{array}$$

$$U^{(3)} = \begin{array}{c} \\ 1\\2\\3\\4\\5\\6\\7 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{bmatrix} 0 & 4 & \underline{5} & 7 & 14 & & \\ & 0 & \underline{6} & 3 & 10 & & \\ & & 0 & 4 & \infty & \underline{9} & \\ & & 3 & 0 & 6 & 3 & \\ & & & 4 & 0 & 3 & 2 \\ & & & & 2 & 0 & 2 \\ & & & & & & 2 \end{bmatrix} \end{array}$$

$$U^{(4)} = \begin{array}{c} \\ 1\\2\\3\\4\\5\\6\\7 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{bmatrix} 0 & 4 & 5 & 7 & 14 & 14 & \\ & 0 & 6 & 3 & 10 & 15 & \\ & & 0 & 4 & \infty & 9 & \\ & & 3 & 0 & 6 & 3 & \\ & & & 4 & 0 & 3 & 2 \\ & & & & 2 & 0 & 2 \\ & & & & & & 0 \end{bmatrix} \end{array}$$

$$U^{(5)} = \begin{array}{c} \\ 1\\2\\3\\4\\5\\6\\7 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{bmatrix} 0 & 4 & 5 & 7 & 13 & 10 & \\ & 0 & 6 & 3 & 9 & 6 & \\ & & 0 & 4 & 10 & 7 & \\ & & 3 & 0 & 6 & 3 & \\ & & 7 & 4 & 0 & 3 & 2 \\ & & & & 2 & 0 & 2 \\ & & & & & & 0 \end{bmatrix} \end{array}$$

$$U^{(6)} = \begin{array}{c} \\ 1\\2\\3\\4\\5\\6\\7 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{bmatrix} 0 & 4 & 5 & 7 & 13 & 10 & 15 \\ & 0 & 6 & 3 & 9 & 6 & 11 \\ & & 0 & 4 & 10 & 7 & 12 \\ & & 3 & 0 & 6 & 3 & 8 \\ & & 7 & 4 & 0 & 3 & 2 \\ & & 9 & 6 & 2 & 0 & 2 \\ & & & & & & 0 \end{bmatrix} \end{array}$$

$$U^{(7)} = U^{(8)} = \begin{array}{c} \\ 1\\2\\3\\4\\5\\6\\7 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{bmatrix} 0 & 4 & 5 & 7 & 12 & 10 & 12 \\ & 0 & 6 & 3 & 8 & 6 & 8 \\ & & 0 & 4 & 9 & 7 & 9 \\ & & 3 & 0 & 5 & 3 & 5 \\ & & 7 & 4 & 0 & 3 & 2 \\ & & 9 & 6 & 2 & 0 & 2 \\ & & & & & & 0 \end{bmatrix} \end{array}$$

223
3

244
44
44

$$P_{ij} = \begin{cases} i \cdot j & : f_{ij}=0\text{のとき} \\ P_{i f_{ij}} \cdot P_{f_{ij} j} & : f_{ij}\neq 0\text{のとき} \end{cases}$$

4

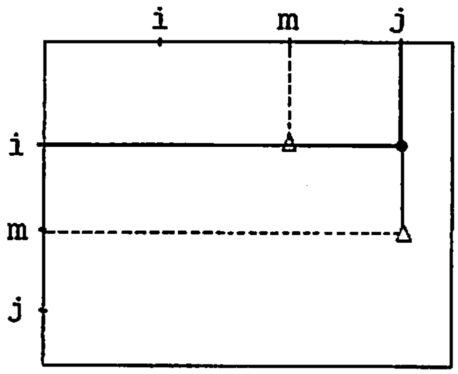
2 4 4 5
4 5
5
5

4
5 5

$$F = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \left[\begin{array}{ccccccc} & & & 2 & 6 & 4 & 6 \\ & & & & 6 & 4 & 6 \\ & & & & 6 & 4 & 6 \\ & & & & 6 & & 6 \\ & & & 4 & & & \\ & & & 5 & 5 & & \\ & & & & & & \end{array} \right] \end{array}$$

$$P_{16} = P_{14} \cdot P_{46} = P_{12} \cdot P_{24} \cdot P_{46} = 1 \cdot 2 \cdot 4 \cdot 6$$

$$u_{ij}^{(m+1)} = \mathrm{MIN}\{\, u_{ij}^{(m)},\ \ u_{im}^{(m)} + u_{mj}^{(m)} \,\}$$



$$u_{23}^{(2)} = \mathrm{MIN}\{\, u_{23}^{(1)}\ ,\ \ u_{21}^{(1)} + u_{13}^{(1)} \}$$

$$= \mathrm{MIN}\{6,\ \ \infty + 5\ \}$$

$$= 6$$

$$u_{14}^{(3)} = \mathrm{MIN}\{\, u_{14}^{(2)}\ ,\ \ u_{12}^{(2)} + u_{24}^{(2)} \}$$

$$= \mathrm{MIN}\{\infty,\ \ 4 + 3\ \}$$

$$= 7$$