

# 情報とコンピュータ

静岡大学工学部  
安藤和敏

2006.01.30

# 11章 プログラム実行時間

計算容易な問題と計算困難な問題

## 検索

	名前		身長		体重
1	John Jones	1	67	1	120
2	Sue Black	2	67	2	131
3	Bill Smith	3	73	3	166
4	Frak Doe	4	68	4	140
5	Jean White	5	67	5	131
6	Nacy Blike	6	71	6	162

## 検索

コンピュータ 目標の身長を与えよ.  
ユーザ 67  
コンピュータ 目標の体重を与えよ.  
ユーザ 131  
コンピュータ 目標の身長・体重の人は以下の通り.  
Sue Black  
Jean White  
検索終了

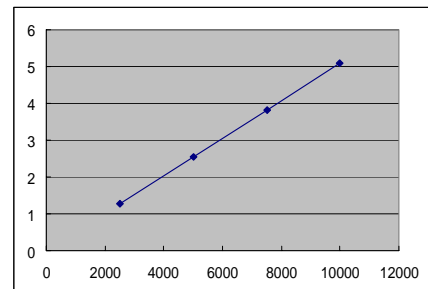
こんなことをやってくれるプログラムを考える。

## 検索プログラムの実行時間

人数 n	実行時間 t(秒)
2500	1.275
5000	2.550
7500	3.825
10000	5.100

$$t = 5.1 \times 10^{-4} \times n$$

## 検索プログラムの実行時間



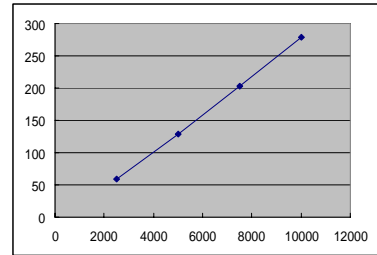
$$t = 5.1 \times 10^{-4} \times n$$

### 先週やったクイックソートの実行時間

n	実行時間 t(秒)
2500	59.261
5000	129.021
7500	202.745
10000	279.042

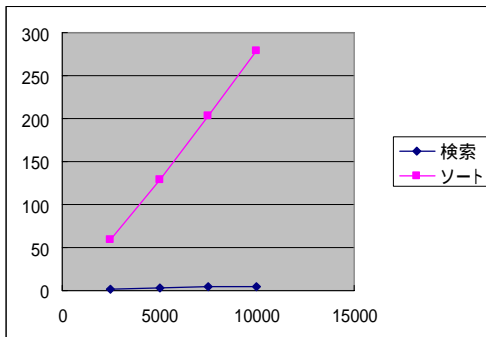
$$t = 2.1 \times 10^{-3} \times n \times \log_2 n$$

### 先週やったクイックソートの実行時間



$$t = 2.1 \times 10^{-3} \times n \times \log_2 n$$

### 検索とソートの時間の比較



### ハノイの塔のプログラムの実行時間

円盤の数 n	実行時間 t(秒)
2500	?
5000	?
7500	?
10000	?

### ハノイの塔のプログラムの実行時間

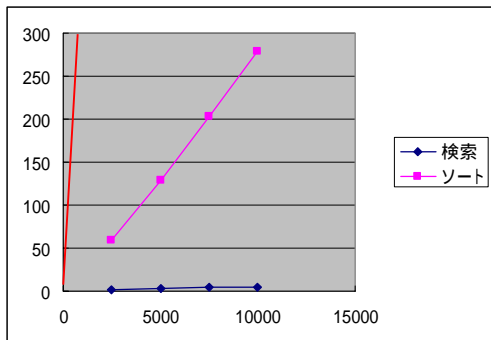
円盤の数 n	実行時間 t(秒)
1	1秒以下
2	1秒以下
3	1秒以下
4	1秒以下
5	1秒以下
6	1秒以下
7	1秒以下
8	1
9	3
10	6

### ハノイの塔のプログラムの実行時間

円盤の数 n	実行時間 t(秒)
11	11
12	23
13	45
14	90
15	180
16	360

$$t = 5.49 \times 10^{-3} \times 2^n$$

### ハノイの塔と検索とソートの時間の比較



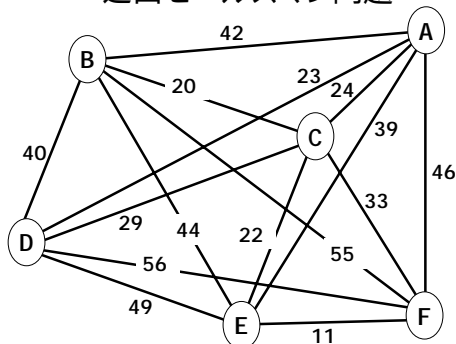
### ハノイの塔のプログラムの実行時間

円盤の数 n	実行時間 t(秒)
5	0.17秒
10	5.62秒
15	3分
20	1.6時間
25	2.13日
30	68.23日
35	5.98年
40	191.30年
45	6120.94年
50	195870年

### ハノイの塔のプログラムの実行時間

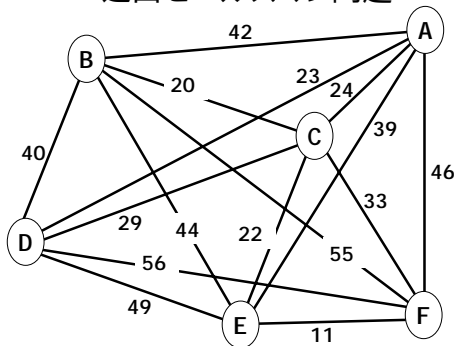
円盤の数 n	実行時間 t(秒)
55	6267840年
60	200571000年
65	6418270000年
70	205385000000年

### 巡回セールスマン問題



Aから出発して全ての都市を1回ずつ訪れて、Aに戻ってくる道筋の中で最短のものは？

### 巡回セールスマン問題



ABCDEF A という道筋の距離は、197

### 巡回セールスマン問題

道筋	距離
AECDBFA	231
ACBDEFA	190
ABFEDCA	210
AEFDCBA	197
AFDECBA	235

上の5つの道筋の中で最短のものは、190である。  
しかし、もっと短い道筋もあるかも知れない。  
道筋の数は、全部で60個ある。

## 巡回セールスマン問題に対する素朴なアルゴリズム

1. 全ての可能な道筋を考える.
2. それらの道筋の距離を計算する.
3. 最短の距離を持つ道筋を出力する.

道筋の数 (都市が6個のとき)

$$5 \times 4 \times 3 \times 2 \times 1 = 5!$$

BCDEF  
の順列の  
数

BCDEFとFEDCBは同じ道筋と  
考えられるから、同じものを  
考えないようにすれば、

$$\text{道筋の数} = 5! / 2$$

道筋の数 (都市がn個のとき)

都市の数が6個のとき  
同じ考えかたで、

$$\text{道筋の数} = (n-1)! / 2$$

$(n-1)!/2$ はどれくらい大きいのか？

n	$(n-1)!/2$
6	60
10	181440
20	60822558204416
50	$3.04 \times 10^{62}$
100	$4.67 \times 10^{155}$

## 50都市の場合の計算時間

仮にコンピュータが1億分の1秒 ( $= 10^{-8}$ 秒)で1つの道筋の距離を調べることができると仮定すると、

$3.04 \times 10^{62}$ 個の道筋(50都市)の全てを調べ終わるまでに、 $3.04 \times 10^{54}$ 秒かかる。

$$3.04 \times 10^{54} \text{秒} = 9.64 \times 10^{54} \text{年}$$

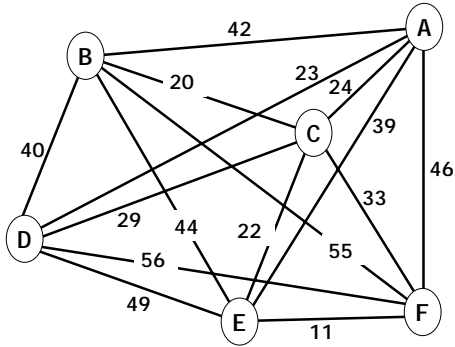
(ちなみに、ビッグバンから現在まで、 $3.15 \times 10^{17}$ 秒 (=100億年)くらいしか経過していない。)

もっといい方法はないのか？

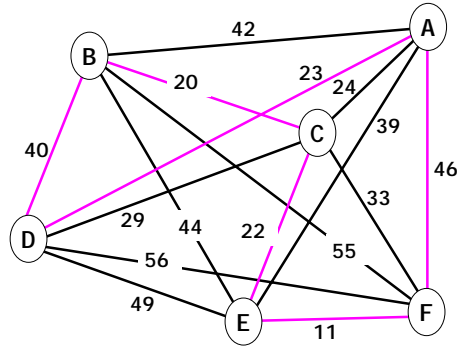
1. 1つの都市を無作為に選ぶ。
2. 現在いる都市から最も近いところにある、まだ訪れていない都市を選択する。まだ、訪れていない都市が残っている限り、このステップを繰り返す。
3. 最後に訪れた都市と最初に訪れた都市とを結ぶ。

このアルゴリズムは速いが、最短の道筋を見つけ出せない。

### 巡回セールスマン問題



### この道筋の方が短い



いい方法はないのか？

現在までのところ誰も「いい方法」を思いついていない。  
もしも、そんな「いい方法」が発見すれば、一大ニュースになるくらいこれは重要な問題である。

たぶん、そんな「いい方法」なんて存在しないだろうと  
みんな思っているのであるが、「いい方法」が存在しない  
ということも証明されていない。