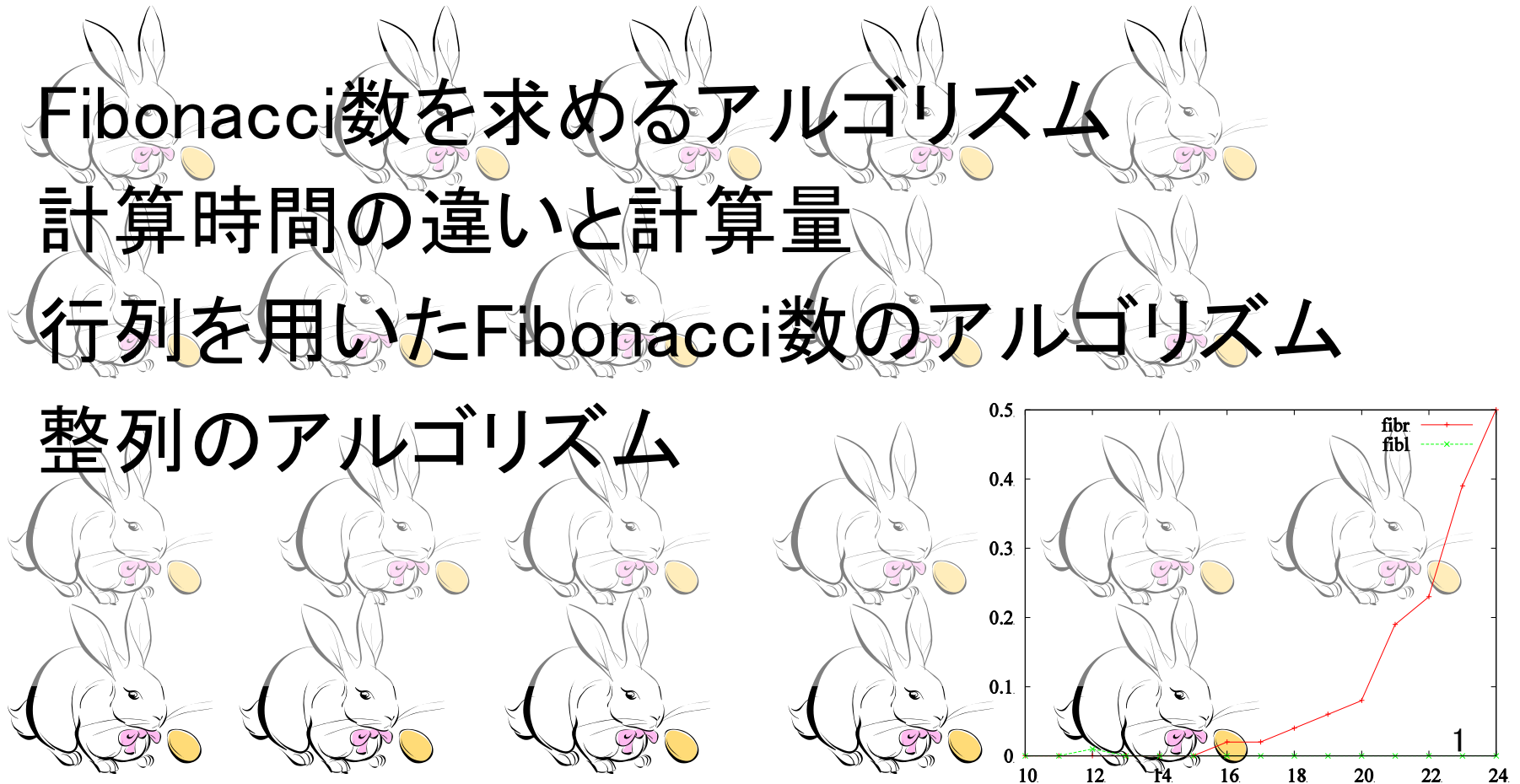


第II部 プログラミングを通して学ぶ 情報科学の諸概念

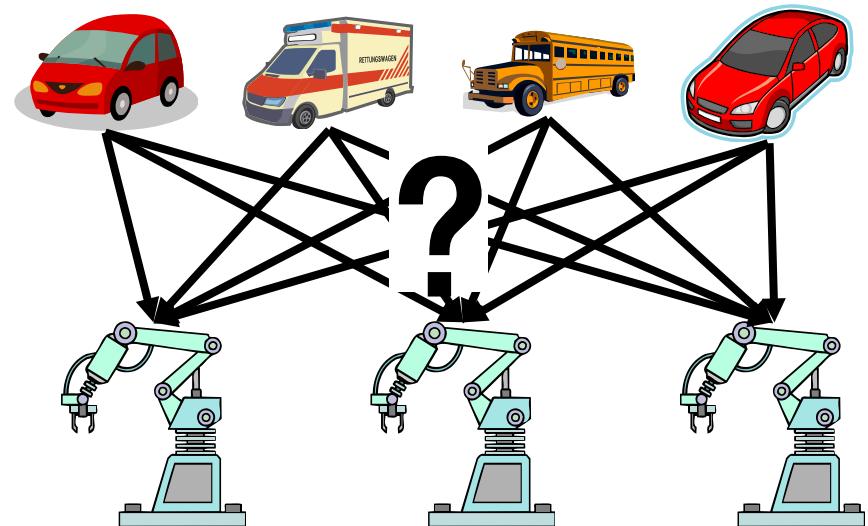
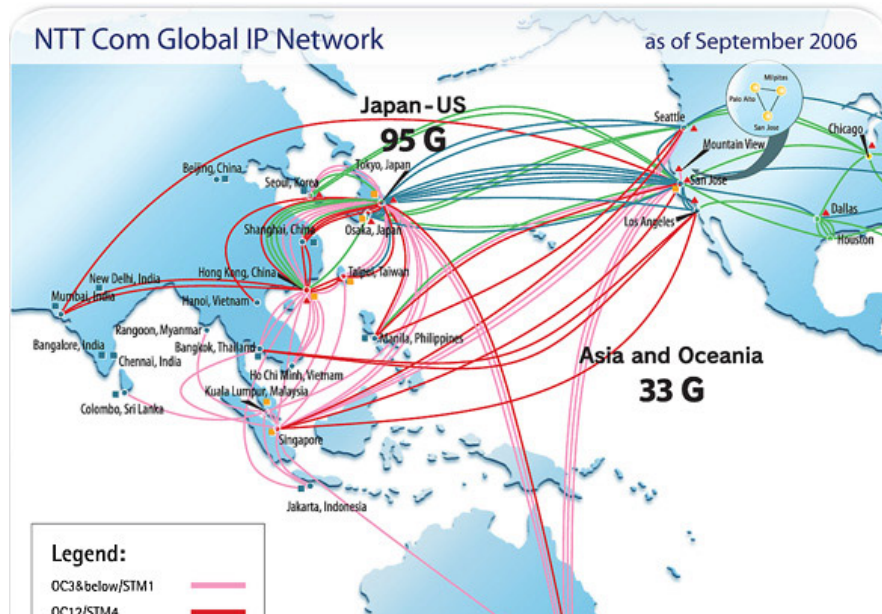
第5章 アルゴリズムと計算量

- ・ Fibonacci数を求めるアルゴリズム
- ・ 計算時間の違いと計算量
- ・ 行列を用いたFibonacci数のアルゴリズム
- ・ 整列のアルゴリズム



アルゴリズム

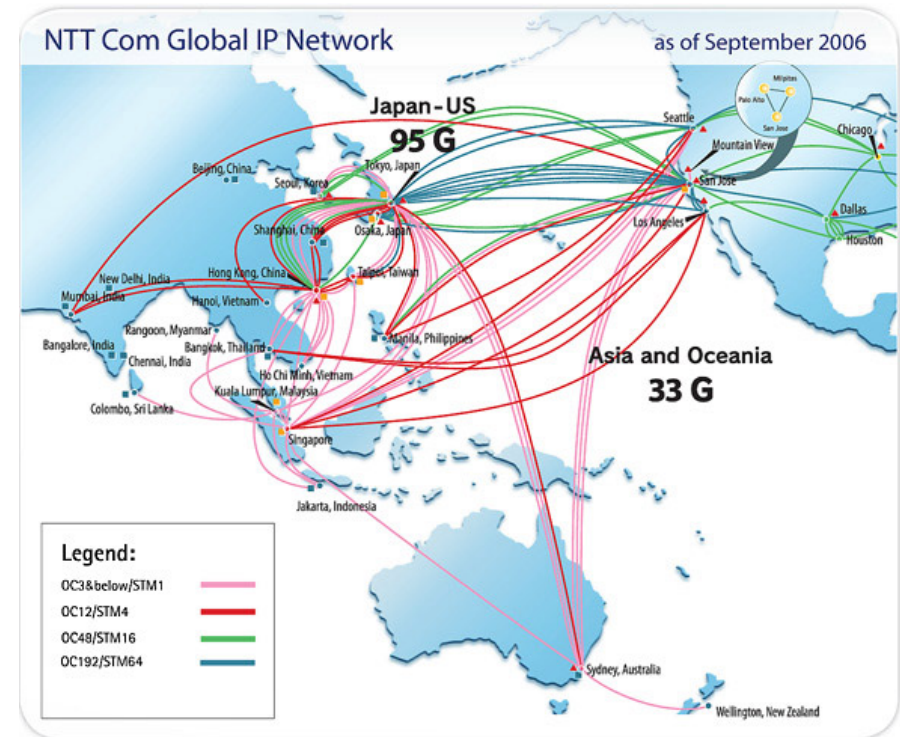
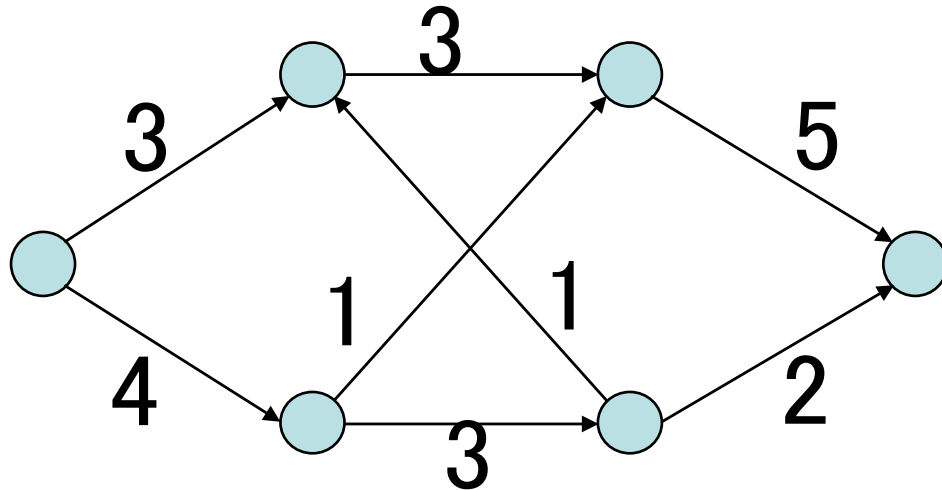
- ・ 問題の解き方
 - 有限時間で答が出ることが保証されている
- ・ 一つの問題に複数のアルゴリズム
 - アルゴリズムによって計算時間が劇的に違う
- ・ 異なる問題に同じアルゴリズム
 - 使いまわしができる



異なる問題、同じアルゴリズム

- ・ 日米間には沢山のネットワーク経路がある
- ・ 各経路には帯域幅がある
- ・ 日米間全体の最大帯域幅は?





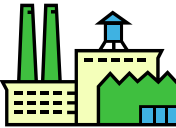

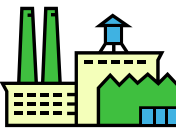
➤ 注: 単純な和でない



異なる問題、同じアルゴリズム

生産割当問題

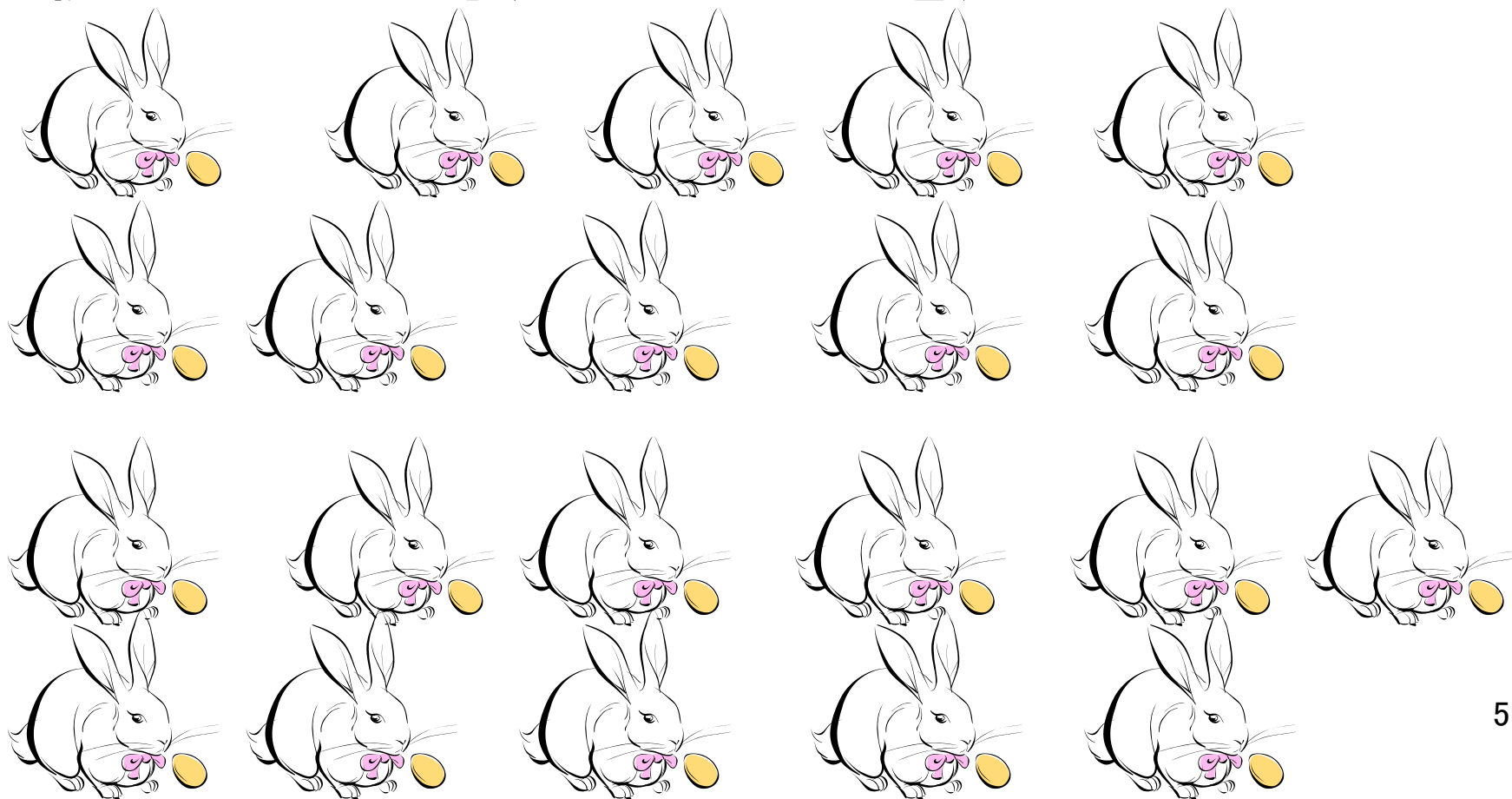
- 車種別の必要台数
- 工場別の生産台数の上限
- どの工場でどの車種を作れるか

	 30	 40	 20	 30
 50			X	
 40		X		
 30	X			X ₄

Fibonacci数

- ・ 子ウサギは1月で親ウサギになる
- ・ 親ウサギは毎月子ウサギを産む

7月: 21羽



Fibonacci数の計算

- ・ 問題: Fibonacci数列の x 番目の数を求める
 - Fibonacci数列: 1, 1, 2, 3, 5, 8, 13, 21, 34, ...
 - k 番目を $\text{fib}(k)$ と書くと $\text{fib}(k) = \text{fib}(k-1) + \text{fib}(k-2)$ となる数列
ただし $\text{fib}(0) = \text{fib}(1) = 1$
- ・ アルゴリズム
 - 再帰的
 - 数え上げ
 - 行列を用いる

Fibonacci数: 再帰的アルゴリズム

$$\text{fib}(k) = \begin{cases} 1 & (k = 0 \text{ または } k = 1) \\ \text{fib}(k - 1) + \text{fib}(k - 2) & (k \geq 2) \end{cases}$$

- ・ 続く2つの値
の和が次の値
になっている

クイズ: fib(100)は

0:偶数 1:奇数

投票のやり方:
ターミナルで

```
cd Downloads  
ruby c.rb 0か1
```

```
def fibr(k)  
  if k==0 || k==1  
    1  
  else  
    fibr(k-1) + fibr(k-2)  
  end  
end  
fibr(100)
```

fib.rb

フィボナッチ数: 数え上げアルゴリズム

フィボナッチ数: 数え上げアルゴリズム

```

def fib1(k)
  f=1
  p1=1
  for i in 2..k
    p2 = p1      #fib(i-2)
    p1 = f      #fib(i-1)
    f = p1 + p2 #fib(i)
  end
  f             #fib(k)
end

```

このとき $f = \text{fib}(i-1)$,
 $p1 = \text{fib}(i-2)$,
 $p2 = \text{fib}(i-3)$

k	0	1	2	3	4
$\text{fib}(k): f$	1	1	2	3	5
1つ前: $p1$	-	1	1	2	3
2つ前: $p2$	-	-	1	1	2

このとき $f = \text{fib}(i)$,
 $p1 = \text{fib}(i-1)$,
 $p2 = \text{fib}(i-2)$

fib.rb

- $\text{fib1}(100)$ の値は?

スピード競争

- 共通資料から、bench.rb と fib.rb をダウンロード。

```
irb(main):004:0> load("./bench.rb")
```

```
irb(main):005:0> load("./fib.rb")
```

```
irb(main):006:0> run("fibr", 10)
```

```
irb(main):007:0> run("fibl", 10)
```

```
irb(main):009:0> for k in 10..24
```

```
irb(main):010:1>   run("fibr", k)
```

```
irb(main):011:1>   run("fibl", k)
```

```
irb(main):012:1> end
```

スピード競争(続き)

```
irb(main):026:0> command("set logscale y")
```

```
irb(main):027:0> for k in 10..32
```

```
irb(main):028:1> run("fibr", k)
```

```
irb(main):029:1> end
```

Y軸を対数スケールに
変更する
(詳しくはテキスト)

- ・ クイズ: fibr(k)の計算時間 t を k の関数で表わすと? (a , b , c は定数)

➤ 1: $t = ak + b$

➤ 2: $t = ak^2 + bk + c$

➤ 3: $t = a \log(k)$

➤ 4: $t = a^k + b$

投票のやり方:
ターミナルで

```
cd Downloads  
ruby c.rb 1~4
```

下 6 桁の Fibonacci 数 (数え上げアルゴリズム)

```
def fib16(k)
  f=1
  p1=1
  for i in 2..k
    p2 = p1
    p1 = f
    f = (p1 + p2) % 1000000
  end
  f
end
```

足し算をするたびに
下6桁だけを残す
(あとは同じ)

```
irb(main):030:0> reset()
```

対数スケールをやめる

```
irb(main):031:0> command("unset logscale")
```

```
irb(main):032:0> for m in 1..10
```

```
irb(main):033:1> run("fib16", 100000*m)
```

10万、20万、30万、
...100万番目の
Fibonacci数

```
irb(main):034:1> end
```

Fibonacci数: 行列を用いたアルゴリズム ($\text{fib}(k+1)$, $\text{fib}(k)$)をベクトルだと思う

$$\begin{pmatrix} \text{fib}(k+1) \\ \text{fib}(k) \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^k \begin{pmatrix} 1 \\ 1 \end{pmatrix} = Q^k \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

- Q^n を高速に計算する方法はあるか? 投票のやり方:
ターミナルで
- $Q^{2n} = (Q^n)^2$ という関係を利用する
➤クイズ: Q^{16} は何回の掛け算でできる?

```
cd Downloads  
ruby c.rb 回数
```

$$Q^n = \begin{cases} E & (n = 0) \\ (Q^{n/2})^2 & (n \text{ は } 2 \text{ 以上の偶数}) \\ Q \times Q^{n-1} & (n \text{ は奇数}) \end{cases}$$

計算量

- ・ Fibonacci数の計算時間を、
関数を定義する前に予想したい
 - 再帰、数え上げ、行列のどれが良い？
- ・ 計算量 = 架空のコンピュータによる計算時間
 - = 基本演算や操作の回数
 - = 実際の計算時間のおおまかな近似を入力の大きさの関数として表わしたもの
 - fib(k)の場合はkが入力の大きさ

Fibonacci数の計算量: 再帰アルゴリズム(1)

isrb中では
どんな関数でも
_CS という
名前を付けると
計算の様子が
表示される

- ・ fibr(5)はどのように計算されるか?

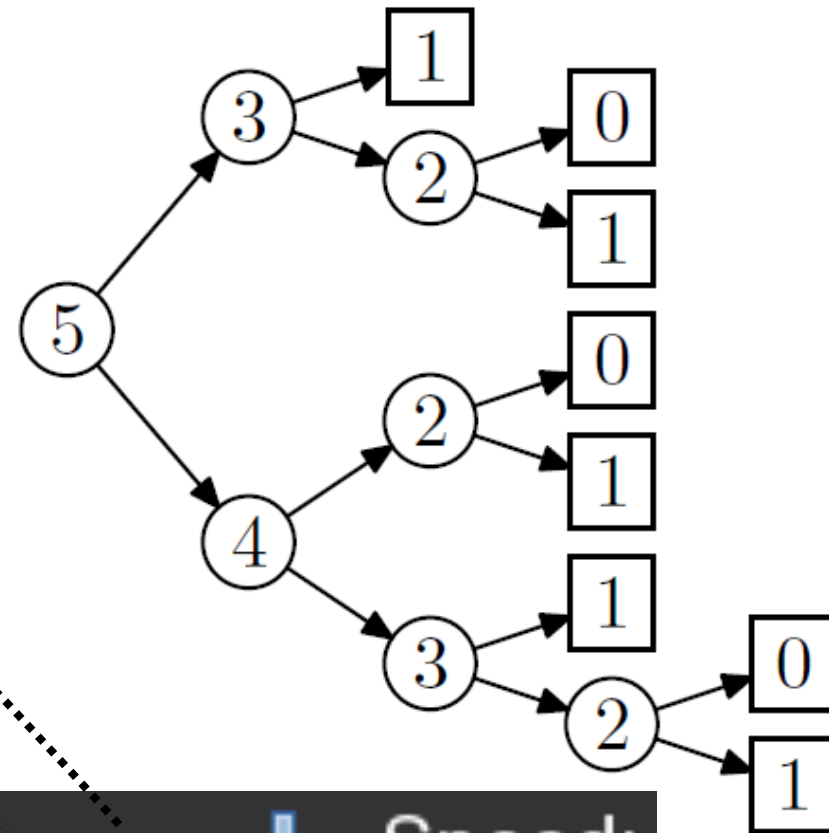
(isrbで)

>> `load("./fib.rb")`

=> true

>> `fibr_CS(5)`

=> 34



ウインドウ左上の「-」を
クリックして“Very
Slow”にしてから再生

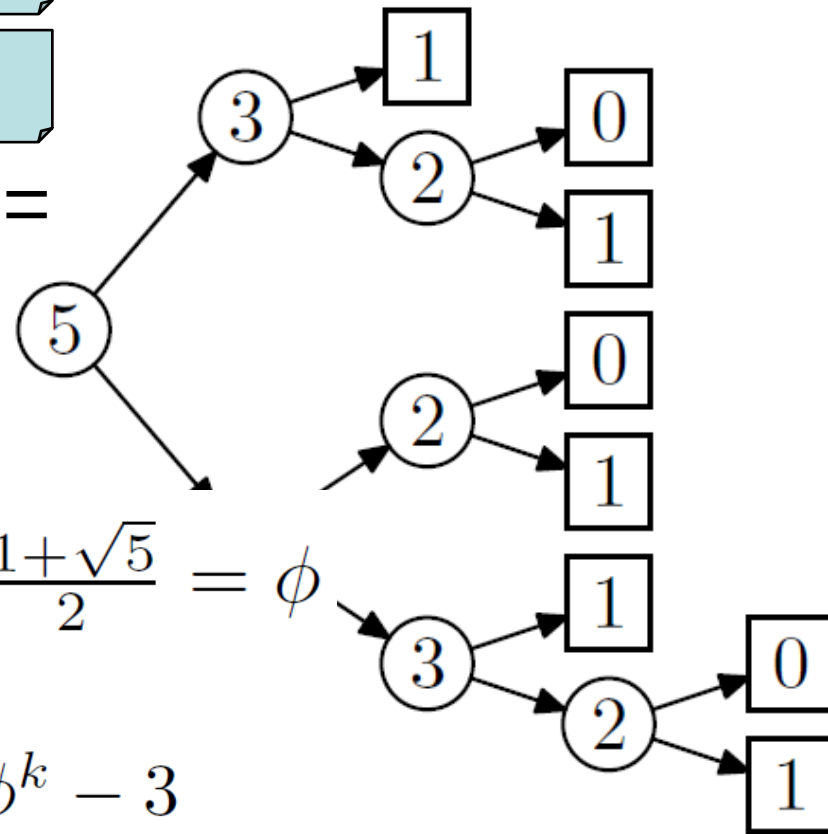


Fibonacci数の計算量: 再帰アルゴリズム(2)

- の数 =
- の数 =
- fibを呼び出した回数 = $2\text{fib}(k)-1$

- $\text{fib}(k) \simeq \frac{\phi^{k+1}}{\sqrt{5}}$ 黄金比 $\frac{1+\sqrt{5}}{2} = \phi$

- fib(k)の計算量 = $\frac{3\phi}{\sqrt{5}}\phi^k - 3$



計算量とO記法

- ・ fibを呼び出した回数 = $2\text{fib}(k)-1$

- ・ $\text{fib}(k) \simeq \frac{\phi^{k+1}}{\sqrt{5}}$

- ・ fib(k)の計算量 = $\frac{3\phi}{\sqrt{5}}\phi^k - 3$

- ・ 計算量: 基本演算や操作の回数を入力の数によって表わしたもの
 - ・ 四則演算、繰り返し、呼び出しなどの回数
 - ・ 最悪の場合を考える
- ・ O(オー)記法: 定数係数や次数の低い項を無視した近似

Fibonacci数の計算量: 数え上げアルゴリズム

k	0	1	2	3	4	5	6	7	8	9	10	11	12
fib(k): f	1	1	2	3	5	8	13	21	34	55	89	144	233
1つ前: p1	—	1	1	2	3	5	8	13	21	34	55	89	144
2つ前: p2	—	—	1	1	2	3	5	8	13	21	34	55	89

- ・ k 回繰り返せばfib(k)が求められる
- ・ 1回あたりの計算は足し算・代入・比較だけ
- ・ 計算量 =
 - 足し算の時間が一定だとした場合

Fibonacci数の計算量: 行列を用いたアルゴリズム

- ・ fib(k)の計算量 = Q^k の計算量
- ・ nは2回に1回は必ず半分以下になる
- ・ 約 $2 \times \log_2 n$ 回の呼び出し
 - 1回あたりの計算: 2×2 行列の乗算ほか
- ・ 計算量 = $O(\log n)$

$$Q^n = \begin{cases} E & (n = 0) \\ (Q^{n/2})^2 & (n \text{ は } 2 \text{ 以上の偶数}) \\ Q \times Q^{n-1} & (n \text{ は奇数}) \end{cases}$$

計算量を求める: Fibonacci数

- 問題: n 番目のFibonacci数

- 定義通り — $O(\psi^n)$ ($\psi=(1+\text{sqrt}[5])/2$)

- 数え上げ — $O(n)$

- 行列 — $O(\log n)$

