

データ構造とアルゴリズム演習 第10回 (12/13)

中嶋 一貴

出席確認

- 出席確認します.
- やむを得ない理由で遅刻・欠席する場合は事前にメールで連絡し、到着後速やかに報告する.
 - その際、確認できる書類（遅延証明書、病院のレシート等）を提出すること.
- 連絡先： nakajima [at] tmu.ac.jp （[at]を@に変える）

動的計画法

- 小さい部分問題の解をメモリに記憶しておいて大きい問題の解を効率的に解くという考え方.
- 同じ計算を繰り返し行うことを回避することで計算を効率化する.

フィボナッチ数列

- 次の漸化式で定義される：

- $F_0 = 0, F_1 = 1, F_{n+2} = F_n + F_{n+1}$

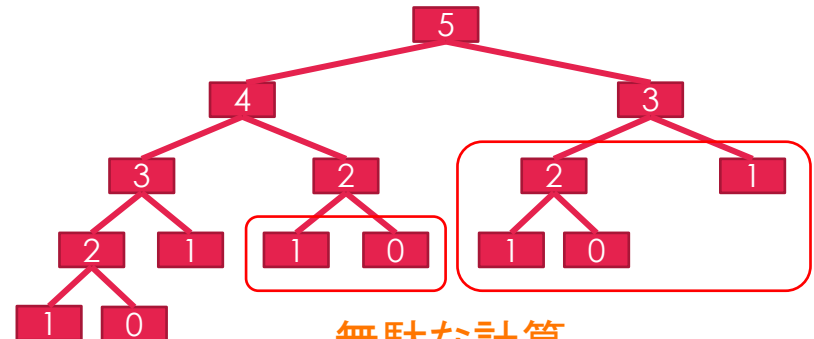
- 数列

- 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 ...

- 素朴な方法：再帰的に計算する.

```
fib(n)
if n == 0 || n == 1:
    return n
return fib(n-1) + fib(n-2)
```

n=5



フィボナッチ数列

- より良い方法：
 - 一度計算したところはメモしておくことで再計算を避ける.
 - 小さい部分問題の解をメモリに記憶.

F ← 各要素を-1で埋めた
十分大きな配列

F[0] ← 0

F[1] ← 1

fib(n):

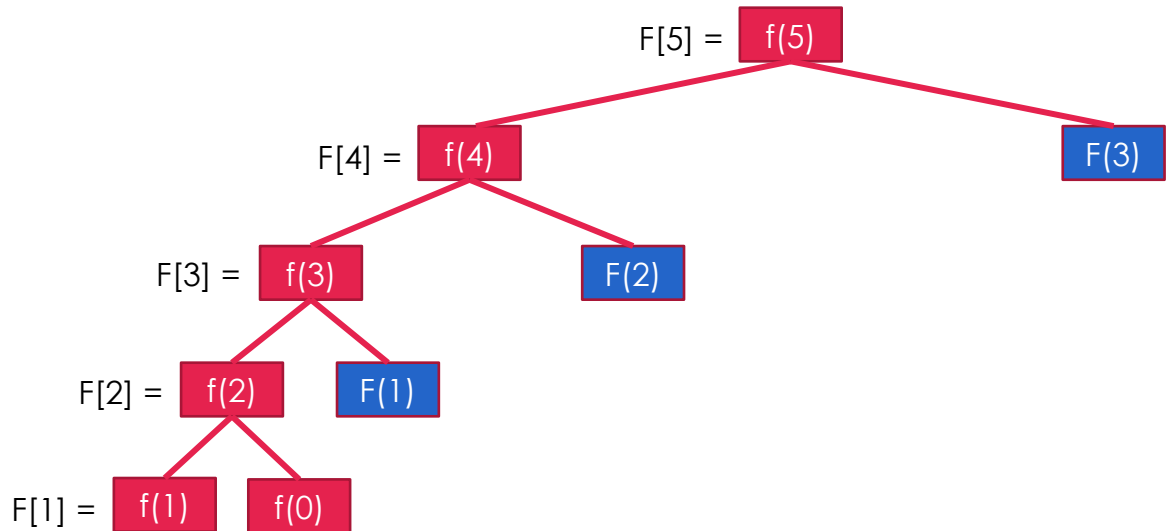
if F[n] != -1:

return F[n]

else:

F[n] = fib(n-1)+fib(n-2)

return F[n]



連鎖行列積

- 行列積は計算順序によって計算量が大きく異なる.
- 行列 M_i は p_{i-1} 行 p_i 列とする.
- 2つの行列の積
 - $(M_1 M_2)$ の計算量は $p_0 p_1 p_2 \rightarrow$ メモしておく.
 - $(M_2 M_3)$ の計算量は $p_1 p_2 p_3 \rightarrow$ メモしておく.
- 3つの行列の積
 - $(M_1 (M_2 M_3))$ の計算量: (M_1) の計算量 + $(M_2 M_3)$ の計算量 + $p_0 p_1 p_3$
 - $((M_1 M_2) M_3)$ の計算量: $(M_1 M_2)$ の計算量 + (M_3) の計算量 + $p_0 p_2 p_3$
 - 小さい方を $M_1 M_2 M_3$ の計算量としてメモしておく.
- さらに大きな問題のコストも既に計算した行列積の計算量を利用して効率的に求めることができる.

提出物（締め切り：2023/12/20 10:00）

1. AJOのユーザIDが入った提出履歴のページのスクリーンショット（全問のステータスが「AC」であること）

- 課題

- ALDS1_10_A : Fibonacci Number
- ALDS1_10_B : Matrix Chain Multiplication

- kibacoの「第10回 課題」から提出すること.

提出物（締め切り：2023/12/20 10:00）

2. 課題を解くために使用したソースファイル一式

- 各ソースコードが、誰のもので、どの課題に対応しているかわかるようにファイル名を設定すること.
- 例えば、氏名がKazuki Nakajima で課題 ALDS1_10_A に対する C++ コードの ファイル名は

alds1_10_a_kazuki_nakajima.cpp

- ソースコードが複数に分かれている場合は、zip ファイルにまとめて提出すること.
- zip にまとめずに複数のソースファイルを提出した方は減点対象.
- kibacoの「第10回 課題」から提出すること.

提出物（締め切り：2023/12/20 10:00）

3. クイズの答案（word ファイル）

- クイズ（quiz_10.docx）の空欄を埋めた word ファイル.
- kibacoの「第10回 課題」から提出すること.

その他注意事項

- 提出期限を過ぎると、kibaco から課題を提出できません.
 - 提出期限後に課題を提出する場合、下記にご連絡ください.
連絡先：nakajima [at] tmu.ac.jp ([at]を@に変える)
- 提出ファイルを間違えていないか、提出前によく確認する.
 - 間違ったファイルを提出されても、いちいち連絡しません.
- 成績が確定するまで、tmuメールをよく確認する.
 - 成績に関わる事項で教員から連絡する場合があります.
- コピペなど不正行為厳禁.