# EDPC解説 (A ~ E)

TAB

# DP（動的計画法）とは？

細かくアルゴリズムが定義されているわけではなく、下記2条件を満たすアルゴリズムの総称である[1]。

- 分割統治法：部分問題を解き、その結果を利用して、問題全体を解く
- メモ化：部分問題の計算結果を再利用する

なお、言葉の定義に下記制約条件を付ける人もいる。

- 最適化問題である
- ボトムアップである（つまり、部分問題を解き終わるまで問題全体に手を出してはいけない）

# A問題：Frog 1

- 問題概要

  1 ~ N 番の足場が順番に並んでいる。i 番目の足場の高さは $h_i$

  i 番の足場からは i + 1 , i + 2 番の足場に移動できる

  i 番の足場から j 番の足場に移動するのにかかるコストは $|h_i - h_j|$

  1 番の足場から N 番の足場まで移動するのにかかるコストの最小値を答えよ

- 制約
  入力はすべて整数である
  $1 \leq N \leq 10^5$
  $1 \leq h_i \leq 10^4$

# A問題：Frog 1

- 足場 i まで行くのにかかるコストの最小値を $dp_i$ とする。
- 足場 i に行くには、i - 1 から行く場合と i - 2 から行く場合がある。
  i - 1 までにコストがいくらかかったかに関わらず、i - 1 から i への移動にかかるコストは一定。( i - 2 についても同様)
- 従って、 $dp_{i-1}$ , $dp_{i-2}$ が分かれば $dp_i$ がわかる。

$$dp_i = \min(dp_{i-1} + |h_i - h_{i-1}|, dp_{i-2} + |h_i - h_{i-2}|)$$

各 $dp_i$ を求めるのは O(1) でできるので、全体で O(N)

# B問題：Frog 2

- 問題概要

  1 ~ N 番の足場が順番に並んでいる。i 番目の足場の高さは $h_i$

  i 番の足場からは i + 1 , i + 2, … i + K 番の足場に移動できる

  i 番の足場から j 番の足場に移動するのにかかるコストは $|h_i - h_j|$

  1 番の足場から N 番の足場まで移動するのにかかるコストの最小値を答えよ

- 制約

  入力はすべて整数である

  $1 \leq N \leq 10^5$
  $1 \leq h_i \leq 10^4$
  $1 \leq K \leq 100$

# B問題：Frog 2

- A問題は二つ先までしか跳べなかったけど、今回は K 個先まで跳べる。
- 足場 i に行く直前は i - 1, i - 2, ..., i - K のどこかにいる。
- 足場 i まで行くのにかかるコストの最小値を $dp_i$ とする。

$$dp_i = \min_{1 \leq j \leq K} \left( dp_{i-j} + \left| h_i - h_{i-j} \right| \right)$$

$dp_i$ を求めるのは O(K) でできる。
全体で O(NK)

# C問題：Vacation

- 問題概要

  夏休みが N 日ある。それぞれの日には以下の行動のうちどれか一つを行う。

  1. 海で泳ぐ。幸福度$a_i$を得る。
  2. 山で虫取りをする。幸福度$b_i$を得る。
  3. 家で宿題をする。幸福度$c_i$を得る。

  二日連続で同じことをすることはできない。
  N 日間で得られる幸福度の最大値をもとめよ。

- 制約
  入力はすべて整数である

  $1 \leq N \leq 10^5$
  $1 \leq a_i, b_i, c_i \leq 10^4$

# C問題：Vacation

- $dp_i$ を i 日目までに得られる幸福度の和の最大値で解ける？
- 解けません。
  二日続けて同じ行動をとってしまう場合があり得る。
- 前の日にどの行動をとったのかさえ分かれば良い。
  $dp_{i,j}$ を i 日目に j をした時の幸福度の和の最大値とすれば良い。
  $$dp_{i,j} = \max_{k\,!=j}(dp_{i-1,k}) + x_i$$
  （$x$　は j に対応した a, b, c のどれか）
  $dp_{i,j}$ は O(1) で求められる。
  全体で O(N)

# D問題：Knapsack1

- 問題概要

1, 2, … N 番の品物がある。
i 番の品物は重さが $w_i$ で価値が $v_i$
N 個の中から重さの和が W を超えないようにいくつかの品物を選ぶ。
価値の総和の最大値を答えよ。

- 制約

入力はすべて整数である。
$1 \leq N \leq 100$
$1 \leq W \leq 10^5$
$1 \leq w_i \leq W$
$1 \leq v_i \leq 10^9$

# D問題：Knapsack1

- i 番目の荷物をとれるかどうかを判断するためには、i 番目までに取った荷物の重さの和を覚えていないといけない。
- $dp_{i,j}$ = i 番目まで見て、取った荷物の重さの和が j の時の価値の最大値とする。

$$dp_{i,j} = \max\left(dp_{i-1,j}, dp_{i-1,j-w_i} + v_i\right)$$

計算量は O(NW)

# E問題：Knapsack2

- 問題概要

1, 2, … N 番の品物がある。
i 番の品物は重さが $w_i$ で価値が $v_i$
N 個の中から重さの和が W を超えないようにいくつかの品物を選ぶ。
価値の総和の最大値を答えよ。

- 制約

入力はすべて整数である。
$1 \leq N \leq 100$
$1 \leq W \leq 10^9$
$1 \leq w_i \leq W$
$1 \leq v_i \leq 10^3$

# E問題：Knapsack2

- さっき と一緒？
- いいえ
  O(NW) は間に合わない。
- D 問題と比べて v が小さいことに注目。
- $dp_{i,j}$ = i 番目までみて取った荷物の価値の総和が j の時の重さの和の最小値として DP をすればよい。
- 計算量は O(N$\sum v$)

# ? 問題：Knapsack

- Knapsack 問題は色々変種がある。（AOJのDPLにいろいろのってる）
  今回扱ったのは 0 - 1 Knapsack と呼ばれるタイプのもの。
  http://judge.u-aizu.ac.jp/onlinejudge/finder.jsp?course=DPL
- （価値が最大となるような選び方は何通りあるか？みたいなのも解ける）

# リンク集

- drken さんの Qiita の記事
  A から E の解説、類題が載っている。実装方針も色々解説している。
- Typical DP Contest
  EDPC より難易度高めの DP まとめコンテスト

# F問題：LCS

- 問題概要

  文字列 s, t が与えられる。
  s の部分列であり、t の部分列でもあるような文字列のうち最長のものをひとつ求めよ。

- 制約

  s, t は英小文字のみからなる文字列である。
  $1 \leq |s|, |t| \leq 3000$

# F問題：LCS

- 共通部分列の長さの最大値を求める DP を考える。
- 更新するときにどこから遷移してきたかを記録しておけば復元できる。
  これは一般的なテクで、今までの問題もこれをやれば復元できる。
  （ナップザック問題で、どの荷物を選べば価値を最大にできるか、など）