4. コイン集め (Coin Collecting)

解説：保坂

JOI 本選 2019/02/10

問題

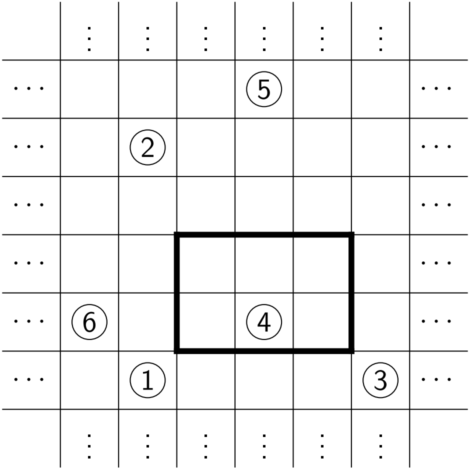
マス目の $(X_1, Y_1), (X_2, Y_2), \ldots, (X_{2N}, Y_{2N})$ にあるコインを $(1,1), (1,2), (2,1), (2,2), \ldots, (N,1), (N,2)$ に 1 枚ずつあるようにするための最小移動回数を求めよ

- 移動は縦横に 1 マスずつ
- 途中でコインが重なってもよい
- 座標は $[-10^9, 10^9]$

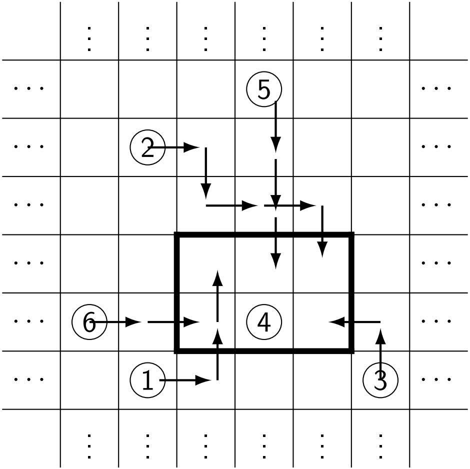
小課題

1. $N \leqq 10$
2. $N \leqq 1\,000$
3. $N \leqq 100\,000$

例



例



単純な解法

どのコインをどのマスにもっていくか決めると，あとはコインごとに独立に動かせばよい

- (X, Y) から (x, y) への最小移動回数は $|x - X| + |y - Y|$
  - Manhattan 距離
- 行き先の決め方は $(2N)!$ 通り
  - $20! > 2 \times 10^{18}$
  - すべて調べるのは小課題 1 でも無理

割り当て問題

動的計画法 (bit DP) で $O(2^{2N}N)$ 時間で解ける (小課題 1)

- 目標マスの部分集合 S に対して，コイン $1, 2, \ldots, |S|$ を S のマス 1 個ずつに移動させるための最小回数を $\mathrm{dp}[S]$ とおく

$O(2^{2N}N)$ 時間解法

foreach $S \subseteq$ (目標マスの集合): // 集合の包含の順に
  foreach $c \in$ (目標マスの集合) $\setminus S$:
    $\mathrm{dp}[S \cup \{c\}] := \min\{\mathrm{dp}[S \cup \{c\}], \mathrm{dp}[S] + ($コイン $|S|+1$ とマス c の距離$)\}$
return dp[目標マスの集合]

考察

- 左右や上下にコインがすれ違うのは無駄
- 左の方のマスには左の方のコインが，右の方のマスには右の方のコインが来そう？

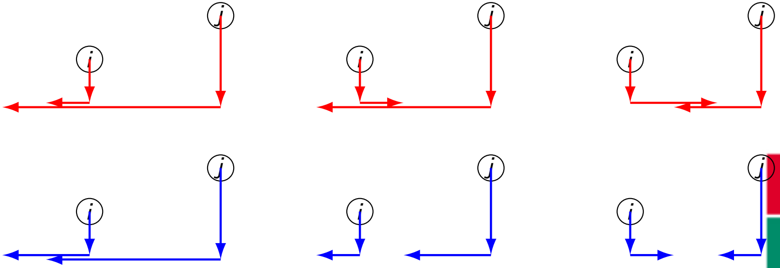
考察

同じ y 座標に行くなら左右を保つ

以下を満たす最適解が存在する：

> コイン i, j の行き先をそれぞれ $(x_i, y_i), (x_j, y_j)$ とするとき，$X_i < X_j$ かつ $y_i = y_j$ ならば，$x_i < x_j$

$x_i > x_j$ なら入れ替えても移動回数が増えないことを示せばよい



2 行に分ける

$y = 1$ にもっていくコインと $y = 2$ にもっていくコインに分けると，あとは左右を保ったまま移動させればよい

上下の分け方をすべて試すと $O(2^{2N} N)$ 時間で解ける

2 行に分ける DP

動的計画法で $O(N^2)$ 時間で解ける (小課題 2)

- コインを X_i の昇順にソートしてから，コイン $1, 2, \ldots, a+b$ を $(1,1), (2,1), \ldots, (a,1), (1,2), (2,2), \ldots, (b,2)$ に 1 個ずつ移動させるための最小回数を $\mathrm{dp}[a][b]$ とおく

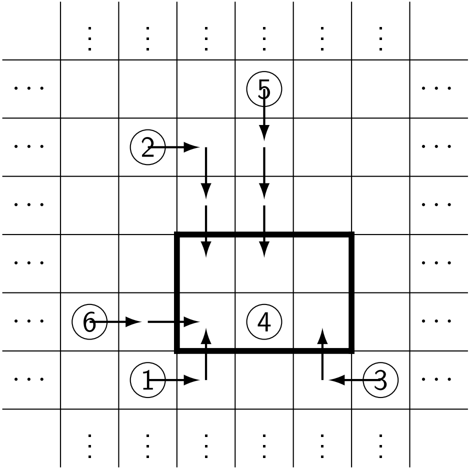
$O(N^2)$ 時間解法

コインを $X_1 \leqq X_2 \leqq \cdots \leqq X_{2N}$ となるようにソート
$\mathrm{dp}[0][0] := 0$
for $a = 0$ **to** N:
  for $b = 0$ **to** N:
    if $a + b < 2N$:
      $\mathrm{dp}[a+1][b] := \min\{\mathrm{dp}[a+1][b],$
        $\mathrm{dp}[a][b] +$(コイン $a+b+1$ と $(a+1, 1)$ の距離)$\}$
      $\mathrm{dp}[a][b+1] := \min\{\mathrm{dp}[a][b+1],$
        $\mathrm{dp}[a][b] +$(コイン $a+b+1$ と $(b+1, 2)$ の距離)$\}$
return $\mathrm{dp}[N][N]$

考察 (集める)

各コインを目標マスのうちの最寄りに移動させてしまってよい

- $1 \leqq X_i \leqq N,\ 1 \leqq Y_i \leqq 2$ として考えられる



1	1	0
2	1	1

枚数

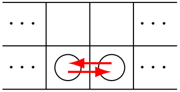
考察 (横移動)

横移動は必要最低限でよい

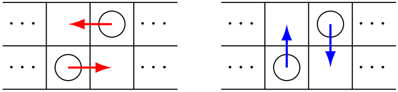
以下を満たす最適解が存在する：

各 $k = 1, 2, \ldots, N-1$ について，$1 \leqq X_i \leqq k$ なるコインの枚数を a_k とおくとき，コインが $x = k$ と $x = k+1$ の間を横に移動する回数は $|a_k - 2k|$

- 左から右へ $a_k - 2k$ 回 or 右から左へ $2k - a_k$ 回は必要
- 左から右へも右から左へも移動しているケースは合計移動回数を増やさずになくせる



これは無駄



縦にしても同じ回数

考察 (横移動)

横移動は必要最低限でよい

以下を満たす最適解が存在する：

各 $k = 1, 2, \ldots, N-1$ について，$1 \leqq X_i \leqq k$ なるコインの枚数を a_k とおくとき，コインが $x = k$ と $x = k+1$ の間を横に移動する回数は $|a_k - 2k|$

例 (入力例 3)

3	1	0	1	2
2	0	0	0	1

$\xrightarrow[3]{}$ $\xrightarrow[2]{}$ $\xleftarrow[1]{}$

考察 (縦移動)

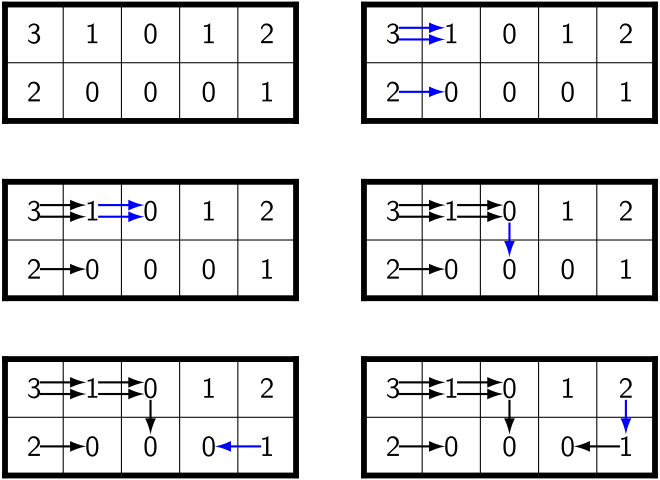
横移動を必要最低限にしたときの縦移動の回数は？

- 縦移動はなるべく右の方で行うことにすれば，縦移動の回数が左から順番に求まる
- あるいは，左から貪欲に必要最低限の縦移動だけ行うと言ってもよい

$y = 1$ ではコインが余っていて $y = 2$ ではコインが足りない，あるいはその逆の状況でのみ，縦移動を行えばよい

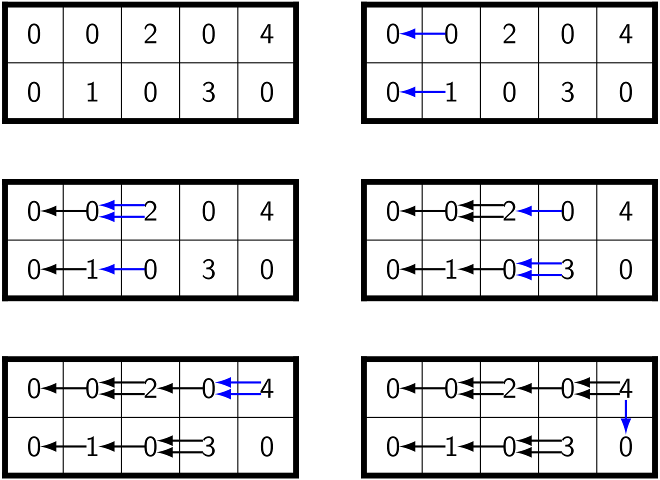
考察 (縦移動)

例 (入力例 3)


3 1 0 1 2
2 0 0 0 1
3 1 0 1 2
2 0 0 0 1
3 1 0 1 2
2 0 0 0 1
3 1 0 1 2
2 0 0 0 1
3 1 0 1 2
2 0 0 0 1
3 1 0 1 2
2 0 0 0 1

考察 (縦移動)

例


0 0 2 0 4
0 1 0 3 0
0 0 2 0 4
0 1 0 3 0
0 0 2 0 4
0 1 0 3 0
0 0 2 0 4
0 1 0 3 0
0 0 2 0 4
0 1 0 3 0
0 0 2 0 4
0 1 0 3 0

まとめ

各箇所の移動回数を順に計算して $O(N)$ 時間で解ける (小課題 3)

- $y = 1, 2$ に対し，(x, y) から $(x+1, y)$ へ横移動するコインの枚数 d_y (負の場合は逆向き) を管理する

$O(N)$ 時間解法

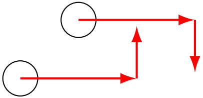
answer := 各コインを最寄りの目標マスに移動させ，そのときの移動回数
$d_1 := 0,\ d_2 := 0$
for $x = 1$ to N:
　$d_1 := d_1 + ((x, 1)$ にあるコインの枚数$) - 1$
　$d_2 := d_2 + ((x, 2)$ にあるコインの枚数$) - 1$
　if $d_1 > 0$ **and** $0 > d_2$:
　　$t := \min\{d_1, -d_2\}$ // $(x, 1)$ から $(x, 2)$ へ t 回縦移動
　　$answer := answer + t,\ d_1 := d_1 - t,\ d_2 := d_2 + t$
　if $d_1 < 0$ **and** $0 < d_2$:
　　$t := \min\{-d_1, d_2\}$ // $(x, 2)$ から $(x, 1)$ へ t 回縦移動
　　$answer := answer + t,\ d_1 := d_1 + t,\ d_2 := d_2 - t$
　$answer := answer + |d_1|$ // $(x, 1)$ と $(x+1, 1)$ の間を $|d_1|$ 回横移動
　$answer := answer + |d_2|$ // $(x, 2)$ と $(x+1, 2)$ の間を $|d_2|$ 回横移動
return *answer*

注意

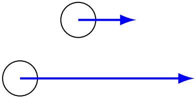
同じ y 座標に行かないときは左右を保つべきとは限らない

1	2	0	1
4	0	0	0

1	2	0	1
4	0	0	0



これはダメ



こっちのほうがよい

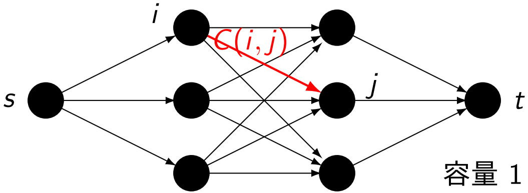
参考：最小費用流

割り当て問題

$n \times n$ のコスト行列 $C(i, j)$ が与えられたとき，$1, 2, \ldots, n$ の並べ替え $p_1, p_2, \ldots, p_n$ であって，$C(1, p_1) + C(2, p_2) + \cdots + C(n, p_n)$ が最小になるものを求めよ

小課題 1 の解説のように $O(2^n n)$ 時間で解けるが，最小費用流あるいは Hungarian algorithm を用いると $O(n^3)$ 時間で解けることが知られている

- 小課題 2 ($n \leq 2\,000$) は無理

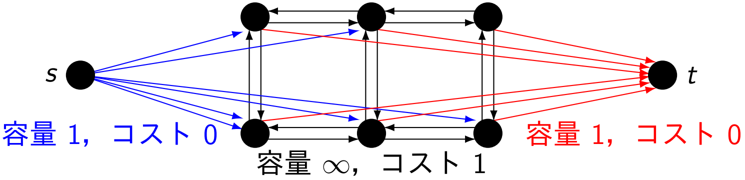


参考：最小費用流

この問題ではコストが特殊なので，グラフの作り方を工夫すると $O(N)$ 頂点 $O(N)$ 辺のグラフでの最小費用流に落とせる

- 最短路反復法で $O(N^2 \log N)$ 時間となり，ある程度高速な実装なら小課題 2 までは通る

1	1	0
2	1	1



得点分布