2019/05/05

17章
動的計画法

M1 mokky

17.1 コイン問題

額面がc_1, c_2, ..., c_m 円のm種類のコインを使って、n円を支払うときの、コインの最小の枚数を求めてください。各額面のコインは何度でも使用することができます。

入力　n m
c_1 c_2 ... c_m

1行目に整数n と整数m が1つの空白区切りで与えられます。2行目に各コインの額面が1つの空白区切りで与えられます。

出力　コインの最小枚数を1行に出力してください。

制約　$1 \leq n \leq 50{,}000$
$1 \leq m \leq 20$
$1 \leq$ 額面 $\leq 10{,}000$
額面はすべて異なり、必ず1を含む。

入力例

```
15  6
1  2  7  8  12  50
```

出力例

```
2
```

17.1.1 解説

d-hacks
JIN NAKAZAWA LAB

貪欲法（greedymethod）

その時点で最適の解（方法）を選んでいくアルゴリズム

1, 5, 50, 100, 500

額面の大きいものから引いて（割って）いけば、最小の枚数を求めることができる

動的計画法

対象となる問題を複数の部分問題に分割し、部分問題の計算結果を
記録しながら解いていく

1, 2, 7, 8, 12, 50

例：15 → 12, 2, 1(貪欲法)
8, 7(動的計画法)

17.1.1 解説

$C[m]$	$C[i]$ を i 番目のコインの額面とする配列
$T[m][n+1]$	$T[i][j]$ を i 番目までのコインを使って j 円支払うときのコインの最小枚数とする2次元配列

コインの枚数 i、各 i における支払う金額 j を増やしていき、
T [i][j]を更新していく。

T [i][j] は、i番目のコインを使わない場合と使う場合の枚数を比べ、
小さい方を選べばよい。

		T	j 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
C			0	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF
0	1		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2		0	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8
i 2	7		0	1	1	2	2	3	3	1	2	2	3	3	4	4	2	3
3	8		0	1	1	2	2	3	3	1	1	2	2	3	3	4	2	2
4	12		0	1	1	2	2	3	3	1	1	2	2	3	1	2	2	2

$$T[j] = min(T[j], T[j - C[i]] + 1)$$

17.1.1 解説

d-hacks
JIN NAKAZAWA LAB

1 << 0	1	0x00000001
1 << 1	2	0x00000002
1 << 2	4	0x00000004
1 << 3	8	0x00000008
1 << 4	16	0x00000010
1 << 5	32	0x00000020
1 << 6	64	0x00000040
1 << 7	128	0x00000080
1 << 8	256	0x00000100
1 << 9	512	0x00000200
1 << 10	1024	0x00000400
1 << 11	2048	0x00000800
1 << 12	4096	0x00001000
1 << 13	8192	0x00002000
1 << 14	16384	0x00004000
1 << 15	32768	0x00008000
1 << 16	65536	0x00010000
1 << 17	131072	0x00020000
1 << 18	262144	0x00040000
1 << 19	524288	0x00080000
1 << 20	1048576	0x00100000
1 << 21	2097152	0x00200000
1 << 22	4194304	0x00400000
1 << 23	8388608	0x00800000
1 << 24	16777216	0x01000000
1 << 25	33554432	0x02000000
1 << 26	67108864	0x04000000
1 << 27	134217728	0x08000000
1 << 28	268435456	0x10000000
1 << 29	536870912	0x20000000
1 << 30	1073741824	0x40000000
1 << 31	2147483648	0x80000000

17.1.2 考察

d-hacks
JIN NAKAZAWA LAB

コイン問題は動的計画法を用いてO(nm) のアルゴリズムで解くことができる

17.2 ナップザック問題

価値がv_i 重さがw_i であるようなN 個の品物と、容量がW のナップザックがあります。次の条件を満たすように、品物を選んでナップザックに入れます：

- 選んだ品物の価値の合計をできるだけ高くする。
- 選んだ品物の重さの総和はW を超えない。

入力　1行目に2つの整数N、W が空白区切りで1行に与えられます。続くN 行でi 番目の品物の価値v_iと重さw_iが空白区切りで与えられます。

出力　価値の合計の最大値を1行に出力してください。

制約　$1 \leq N \leq 100$
$1 \leq v_i \leq 1{,}000$
$1 \leq w_i \leq 1{,}000$
$1 \leq W \leq 10{,}000$

入力例

```
4 5
4 2
5 2
2 1
8 3
```

出力例

```
13
```

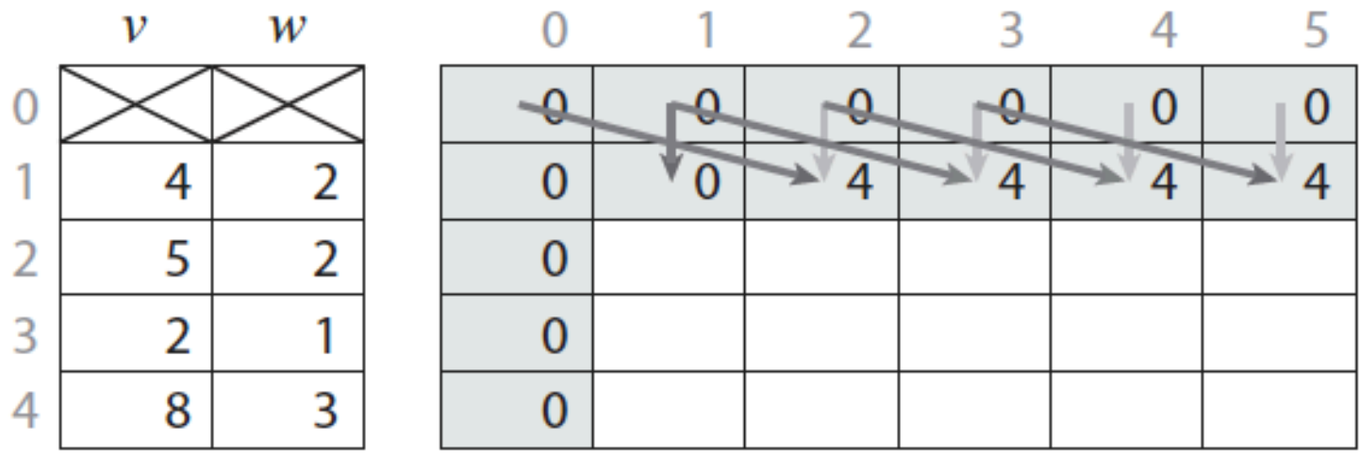
17.2.1 解説

item[N+1]

	v	w
0		
1	4	2
2	5	2
3	2	1
4	8	3

C[N+1][W+1]

	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0					
2	0					
3	0					
4	0					



17.2.1 解説

d-hacks
JIN NAKAZAWA LAB

	v	w
0		
1	4	2
2	5	2
3	2	1
4	8	3

	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	4	4	4	4
2	0	0	5	5	9	9
3	0					
4	0					

	v	w
0		
1	4	2
2	5	2
3	2	1
4	8	3

	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	4	4	4	4
2	0	0	5	5	9	9
3	0	2	5	7	9	11
4	0					

17.2.1 解説

	v	w
0		
1	4	2
2	5	2
3	2	1
4	8	3

	0	1	2	3	4	5
	0	0	0	0	0	0
	0	0	4	4	4	4
	0	0	5	5	9	9
	0	2	5	7	9	11
	0	2	5	8	10	13

	v	w
0		
1	4	2
2	5	2
3	2	1
4	8	3

	0	1	2	3	4	5
	0	0	0	0	0	0
	0	0	4	4	4	4
	0	0	5	5	9	9
	0	2	5	7	9	11
	0	2	5	8	10	13

17.3 最長増加部分列

数列$A = a_0, a_1, ..., a_{n-1}$の最長増加部分列（LIS: Longest Increasing Subsequence）の長さを求めてください。数例Aの増加部分列は$0 \leq i_0 < i_1 < ... < i_k < n$かつ$a_{i_0} < a_{i_1} < ... < a_{i_k}$を満たす部分列$a_{i_0}, a_{i_1}, ..., a_{i_k}$です。最長増加部分列はその中で最も$k$が大きいものです。

入力　1行目に数列Aの長さを示す整数nが与えられます。続くn行で数列の各要素a_iが与えられます。

出力　最長増加部分列の長さを1行に出力してください。

制約　$1 \leq n \leq 100,000$
$0 \leq a_i \leq 10^9$

入力例

```
5
5
1
3
2
4
```

出力例

```
3
```

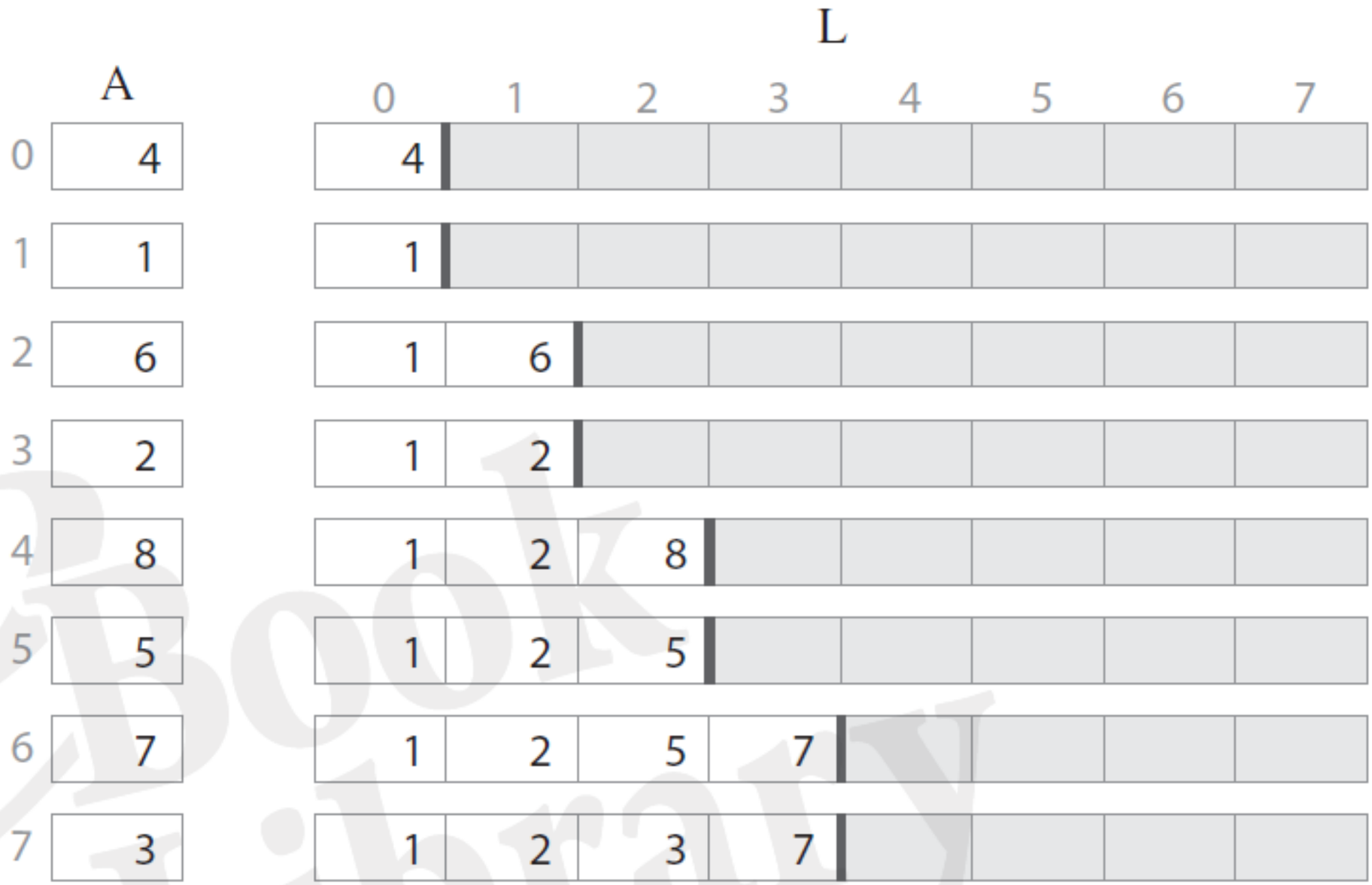
17.3.1 解説

$L[n+1]$	$L[i]$ を $A[1]$ から $A[i]$ までの要素を使い $A[i]$ を最後に選んだときのLISの長さとする配列
$P[n+1]$	$P[i]$ を $A[1]$ から $A[i]$ までの要素を使い $A[i]$ を最後に選んだときのLISの最後から2番目の要素の位置とする配列（得られる最長増加部分列における各要素の1つ前の要素の場所を記録）

	0	1	2	3	4	5	6	7	8
A	-1	4	1	6	2	8	5	7	3
L	0	1	1	2	2	3	3	4	3
P	-1	0	0	1	2	3	4	6	4

計算量は $O(n^2)$

17.3.1 解説



計算量はO(nlogn)

17.4 最大正方形

図のように、一辺が1cm のタイルが、$H \times W$ 個並べられています。タイルは汚れているもの、綺麗なもののいずれかです。

綺麗なタイルのみを使ってできる正方形の面積の最大値を求めてください。

入力　H W

$c_{1,1}$ $c_{1,2}$ …, $c_{1,W}$

$c_{2,1}$ $c_{2,2}$ …, $c_{2,W}$

:

$c_{H,1}$ $c_{H,2}$ …, $c_{H,W}$

1行目に2つの整数H、W が空白区切りで与えられます。続くH 行でタイルを表す$H \times W$ 個の整数c_{ij} が与えられます。c_{ij} が1のとき汚れたタイル、0 のとき綺麗なタイルを表します。

出力　面積の最大値を1行に出力してください。

制約　$1 \leq H, W \leq 1{,}400$

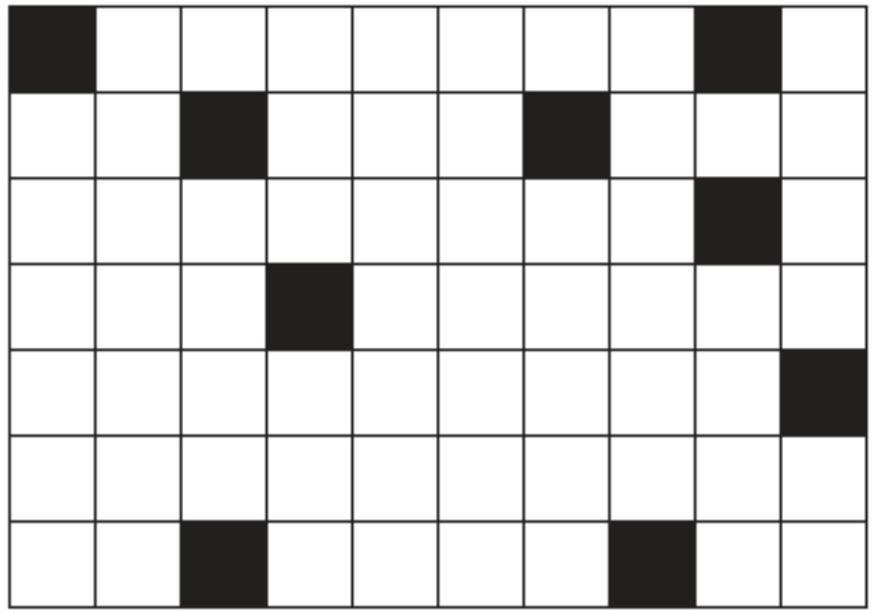
入力例

```
4 5
0 0 1 0 0
1 0 0 0 0
0 0 0 1 0
0 0 0 1 0
```

出力例

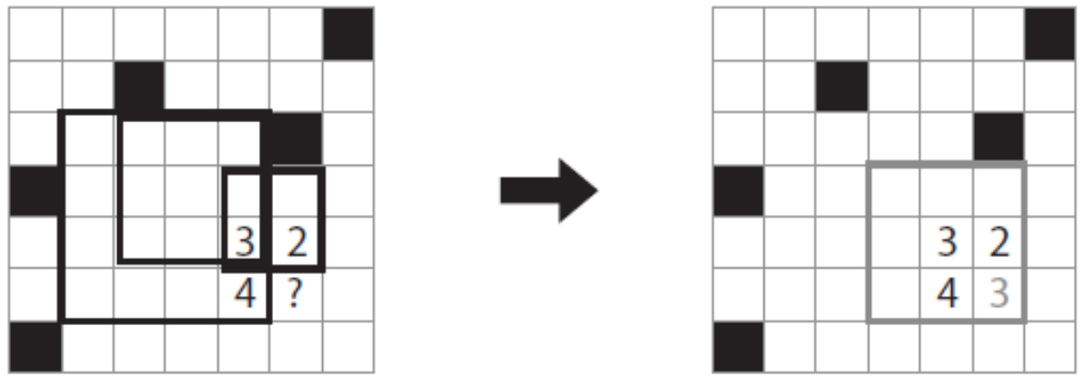
```
4
```

17.4 最大正方形



17.4.1 解説

この問題は次のような動的計画法によって$O(HW)$で解くことができます。小さな部分問題の解を記録するための記憶領域（変数）をdp[H][W]とし、dp[i][j]にタイル(i, j)から左上に向かってできる最大の正方形の辺の長さ（タイルの数）を記録していきます。例えば次のような局面を考えてみましょう。



dp[i][j] の値はその左上、上、左の要素の中で最も小さい値に1 を加えたものになる

17.5 最大長方形

図のように、一辺が1cm のタイルが、$H \times W$ 個並べられています。タイルは汚れているもの、綺麗なもののいずれかです。

綺麗なタイルのみを使ってできる長方形の面積の最大値を求めてください。

入力　H W
$c_{1,1}$ $c_{1,2}$ ..., $c_{1,W}$
$c_{2,1}$ $c_{2,2}$ ..., $c_{2,W}$
:
$c_{H,1}$ $c_{H,2}$ ..., $c_{H,W}$

1行目に2つの整数H、W が空白区切りで与えられます。続くH 行でタイルを表す$H \times W$ 個の整数c_{ij} が与えられます。c_{ij} が1 のとき汚れたタイル、0 のとき綺麗なタイルを表します。

出力　面積の最大値を1行に出力してください。

制約　$1 \leq H, W \leq 1{,}400$

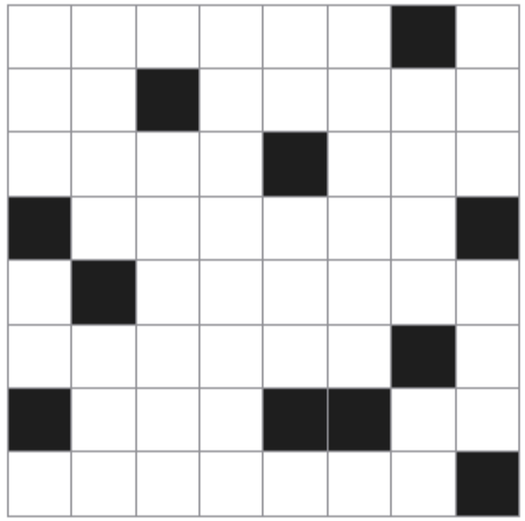
入力例

```
4 5
0 0 1 0 0
1 0 0 0 0
0 0 0 1 0
0 0 0 1 0
```

出力例

```
6
```

17.5.1 解説



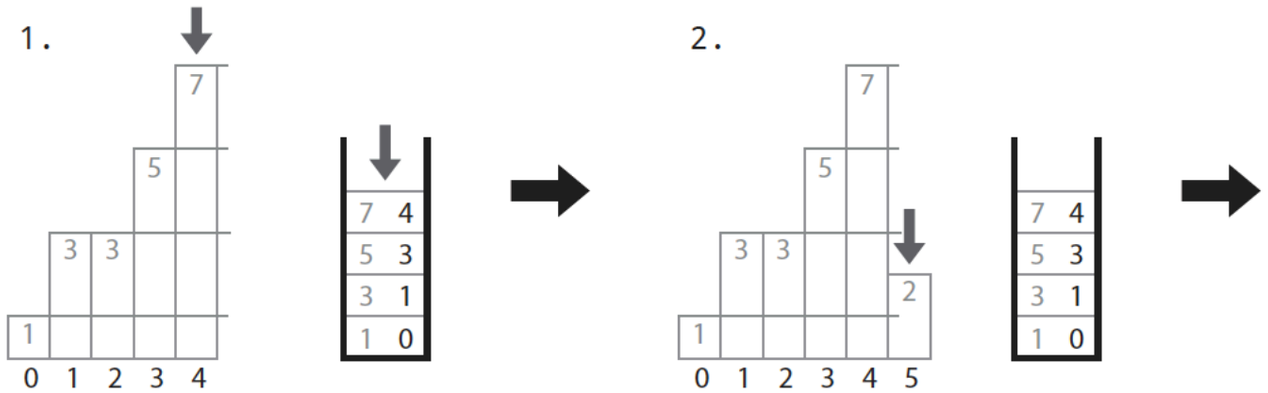
1	1	1	1	1	1	0	1
2	2	0	2	2	2	1	2
3	3	1	3	0	3	2	3
0	4	2	4	1	4	3	0
1	0	3	5	2	5	4	1
2	1	4	6	3	6	0	2
0	2	5	7	0	0	1	3
1	3	6	8	1	1	2	0

各行を１つのヒストグラムとみなす

17.5.1 解説

d-hacks
JIN NAKAZAWA LAB

ヒストグラムが{1, 3, 3, 5, 7, 2}



17.5.2 考察

各ヒストグラムについて、スタックに長方形を追加・削除する操作はO(W)になる。
長方形探索問題では、この処理を各行に行い、O(HW)のアルゴリズムで解くことができる。

17.6 その他の問題

- DPL_1_C: Knapsack Problem
  0-1 Knapsack Problem では、各品物を選ぶか選ばないかの組み合わせでしたが、この問題は各品物をいくつでも選べるナップザック問題です。
- DPL_1_E: Edit Distance (Levenshtein Distance)
  ２つの文字列の編集距離を求める問題です。1 文字の挿入・削除・置換を行って別の文字列に変換する手順の最小回数を動的計画法で求めます。
- DPL_2_A: Traveling Salesman Problem
  巡回セールスマン問題と言われる問題です。重み付きグラフのある頂点から出発し、各頂点をちょうど一度通って出発点へ戻る閉路の最短距離を求める問題です。頂点の数が少ないので、ビット DP と言われるテクニックで解くことができます。
- DPL_2_B: Chinese Postman Problem
  中国人郵便配達問題と言われる問題です。重み付きグラフのある頂点から出発し、各辺を少なくとも一度は通って出発点へ戻る閉路の最短距離を求める問題です。頂点の数が少ないので、ワーシャルフロイドのアルゴリズムとビット DP を使って解くことができます。