Lab10 Solution

YAO ZHAO

Lab10.A Shopping

- Lanran likes shopping! There are n items in the shop, where each one has beauty w_i and costs c_i coins. Lanran has m coins, and he wants to get the largest sum of beauty on items he can buy. Note that, Lanran can buy at most one per item.

Sample input：

```
3 6
5 3 3
3 2 4
```

items	W_i	C_i
1	5	3
2	3	2
3	3	4

items	$sum(C_i)$	$sum(W_i)$
1	3	5
2	2	3
3	4	3
1+2	3+2≤6	5+3 = 8
2+3	3+3≤6	2+4 = 6
1+3	3+4>6	--
1+2+3	3+2+4>6	--

Sample Output：

$\boxed{8}$

Review:

W = 11

#	value	weight
1	1	1
2	6	2
3	18	5
4	22	6
5	28	7

W + 1

n + 1

	0	1	2	3	4	5	6	7	8	9	10	11
ϕ	0	0	0	0	0	0	0	0	0	0	0	0
{ 1 }	0	1	1	1	1	1	1	1	1	1	1	1
{ 1, 2 }	0	1	6	7	7	7	7	7	7	7	7	7
{ 1, 2, 3 }	0	1	6	7	7	18	19	24	25	25	25	25
{ 1, 2, 3, 4 }	0	1	6	7	7	18	22	24	28	29	29	40
{ 1, 2, 3, 4, 5 }	0	1	6	7	7	18	22	28	29	34	34	40

Pseudo-code

```
Input: n, W, w_1,…,w_N, v_1,…,v_N

for w = 0 to W
   M[0, w] = 0

for i = 1 to n
   for w = 1 to W
      if (w_i > w)
         M[i, w] = M[i-1, w]
      else
         M[i, w] = max {M[i-1, w], v_i + M[i-1, w-w_i ]}

return M[n, W]
```

Space complexity optimization of 0-1 knapsack

W + 1

n + 1

	0	1	2	3	4	5	6	7	8	9	10	11
ϕ	0	0	0	0	0	0	0	0	0	0	0	0
{ 1 }	0	1	1	1	1	1	1	1	1	1	1	1
{ 1, 2 }	0	1	6	7	7	7	7	7	7	7	7	7
{ 1, 2, 3 }	0	1	6	7	7	18	19	24	25	25	25	25
{ 1, 2, 3, 4 }	0	1	6	7	7	18	22	24	28	29	29	40
{ 1, 2, 3, 4, 5 }	0	1	6	7	7	18	22	28	29	34	34	40

To calculate the new values of row i, only rows i -1 needed. The data from rows 0 to i -2 do not need storage space.

For W to w_i
    M[w] = max {M[w], v_i +M[w- w_i]}

For W to w_i
    M[i, w] = max {M[i,w], v_i +M[i, w- w_i]}

```
Input: n, W, w1,…,wN, v1,…,vN

for w = 0 to W
   M[0, w] = 0

for i = 1 to n
   for w = 1 to W
      if (wi > w)
         M[i, w] = M[i-1, w]
      else
         M[i, w] = max {M[i-1, w], vi + M[i-1, w-wi ]}

return M[n, W]
```

Lab10.B Game

- Bob is very angry because Alice broke his wonderful TV, so he will battle with Alice in a game to avenge.
- The game is a very simple game. Initially, they are given a non-empty string s, consisting of lowercase letters. The length of the string is even. Each player also has its own empty string. In one move, a player takes either the first or the last letter of the string s, removes it from s and **appends** it to their own string (put it at the end of the string).
- The game ends when s is empty. And then Alice and Bob will compare their strings, the one that owns a lexicographically smaller string will be the winner.
- Bob is very confident with the game, so he will let **Alice move first**. Also, Alice and Bob will take their moves optimally.
- Your task is to tell who is the winner or they draw.

Sample Input:

n: the number of testcases

```
3
ilikealgorithm
ggggggggggg
oooooooooooohhhhhhhhhhh
```

the non-empty string in a game

`ilikealgorithm` Alice move first, if Alice get character **i** first, Bob must lose the game

`ggggggggggg` All characters are the same, will be draw.

`oooooooooooohhhhhhhhhhh`

Alice move first, if Alice get i first, Bob must lose the game

Sample Output:

```
Alice
Draw
Alice
```

only 2 characters:

ij $S_1 < S_2$ Alice is clever $\longrightarrow$ Alice get S_1 $\longrightarrow$ Alice win

ji $S_1 > S_2$ Alice is clever $\longrightarrow$ Alice get S_2 $\longrightarrow$ Alice win

ii $S_1 = S_2$ Draw

4 characters:

ijkl $S_1 < S_4$ $S_1 < S_2$ Alice get $S_1 \longrightarrow$ Bob get S_2 or S_4 $\longrightarrow$ Alice win

jikl $S_1 < S_4$ $S_1 > S_2$ Alice get $S_1 \longrightarrow$ Bob get S_2 $\longrightarrow$ Bob win

Alice get $S_4 \longrightarrow$ Bob get S_1 $\longrightarrow$ Bob win

jlki $S_1 > S_4$ $S_3 > S_4$ Alice get $S_4 \longrightarrow$ Bob get S_1 or S_3 $\longrightarrow$ Alice win

jikl $S_1 > S_4$ $S_3 < S_4$ Alice get $S_4 \longrightarrow$ Bob get S_3 $\longrightarrow$ Bob win

Alice get $S_1 \longrightarrow$ Bob get S_4 $\longrightarrow$ Bob win

ijki $S_1 = S_4 < S_2, S_3$ Alice get $S_1 \longrightarrow$ Bob get S_2 or S_4 $\longrightarrow$ Alice win

Alice get $S_4 \longrightarrow$ Bob get S_1 or S_3 $\longrightarrow$ Alice win

jikj $S_2 < S_1 = S_4 < S_3$ Alice get $S_4 \longrightarrow$ Bob get $S_1 \longrightarrow$ only 2 characters $\longrightarrow$ Alice win

jkij $S_3 < S_1 = S_4 < S_2$ Alice get $S_1 \longrightarrow$ Bob get $S_4 \longrightarrow$ only 2 characters $\longrightarrow$ Alice win kijk $S_2 <= S_3 < S_1 = S_4$ Alice get S_4 → Bob get S_3 → Bob win

Alice get S_1 → Bob get S_2 → Bob win

jjij $S_3 < S_1 = S_4 = S_2$ √ Alice get S_1 → Bob get S_4 or S_2 → only 2 characters → Alice win

Alice get S_4 → Bob get S_3 → Bob win

jjkj $S_1 = S_4 = S_2 < S_3$ Alice get S_1 → Bob get S_4 or S_2 → only 2 characters → Alice win

Alice get S_4 → Bob get S_2 → only 2 characters → Alice win

jjjj $S_1 = S_2 = S_3 = S_4$ Alice get S_1 → Bob get S_4 or S_2 → only 2 characters → Draw

More characters?