主題 | 日期 | 星期

Algorithm Analysis and Design HW-6

6.6. We need to consider both the order of the words and the slacks remained.

So consider an two dimensional array opt[i][j], which records the total slacks by the end of consideration of W_i, remaining a slack as long as j at the end of the last line.

And we have the state transition equation:

If $j \le W_{i+1}$, we can only put W_{i+1} into a new line

$$opt[i][j] \xrightarrow{+L-W_{i+1}} opt[i+1][L-W_{i+1}]$$

Otherwise, we can either draw a new line or stick to this line.

$$opt[i][j] \xrightarrow{-1-W_{i+1}} opt[i+1][j-W_{i+1}-1]$$
$$\xrightarrow{+L-W_{i+1}} opt[i+1][L-W_{i+1}]$$

By this means we can get the algorithm.

opt[0][L] = L (Initial State)

For i = 1, 2, ⋯, n

$$opt[i][L-W_i] = \min_{0\le k<L} opt[i-1][k] + (L-W_i)$$

For $j = L-1-W_i, L-2-W_i, \cdots, 0$

$$opt[i][j] = opt[i][j+W_i+1] - W_i - 1$$

Endfor

主題

日期

星期

Endfor

68. (a) Consider such example (3 days in total):

	1	2	3
x_i	100	100	100
$f(i)$	10	11	12

Obviously we'll never reach the time when $f(j) \geq x_n$
So according to the greedy algorithm we only fight the robots on day 3, which caused 12 death
However the best choice is to beat every day, we can achieve a total death of 30.

(b) Let $opt[i][j]$ be the largest number of the robots beaten when i days past and on day i we beat the robots with $f(j)$ attack

Then obviously we have $opt[i][j] = f(j) + \max_{1 \leq k \leq i-j} opt[i-j][k]$

And the answer to this problem is

$$\max_{1 \leq i \leq n} opt[n][i]$$

主題

日期

星期

6.20. Let opt[i][j] be the total grades of the first i classes when you take j hours on course 1 to course i

Then we have $opt[i][j] = \max\limits_{1 \le t \le j-1} opt[i-1][t]$,

Our answer is $opt[n][H]$

6.27. Let opt[i][j] be the total cost in first i days, while at the end of day i the gas station holds j gallons of oil.

It's clear that each time we import oil, the amount can be a sum of several continuous g_i.

So it's our assumption to import this amount of oil.

On day i $(1 \le i \le n)$, we can choose either import or not.

As a result: $opt[i][j] = \min\{opt[i-1][j+g_i] + (j+g_i)\cdot c,$

$\min\limits_{1 \le t \le j+g_i} \{opt[i-1][j+g_i-t] + P + (j+g_i-t)\cdot c\}\}$

We start from $opt[0][0]=0$ and end when $opt[n][0]$. And $opt[n][0]$ is our final answer.