

HW #4. Dynamic Programming (Binary Input)

(Due: 2017/5/12)

Objective

Practices on way to find the optimal solution.

Introduction

“Shadowverse” (a game similar as “Hearthstone”) is trading card game that players should use their own cards to attack their opponents. Players have limited play points each turn, and all cards will consume different number of play points when they have been played. In this homework you are asked to find the maximum damage by using cards and play points you have.

Rules

There are some rules you should know before you play the game:

1. There are two types of cards, “*follower*” and “*spell*”. A follower will stay on the stage when it has been played, and then you can use followers on the stage to attack your opponents. Unlike follower, a spell will disappear after being played, but it can cause direct damage to your opponents.
2. Each card has 4 attributes, *type*, *cost*, *attack*, and *defense*. Cost means how many play points will consume when this card is played. You cannot play cards if you don’t have enough play points. Attack means the damage of this card can cause. Defense means the damage of this card can bear. Since spells won’t stay on the stage, the defense of all spells are zero.
3. You have an evolution point to evolve one of followers on your stage. After evolved, the attack of this follower will add two.
4. You can only have 5 followers on your stage at the most.

input.txt

Please noted that the input is given in binary format, you are expected to **DECODE THE BINARY FORMAT OF INPUT USING MSGPACK**.

The format of decoded input looks like below:

$\{N$: an **integer-type object** indicating the number of test cases in the input file},
 $\{C_1$: an **integer-type object** indicating the number of cards in the first test case},
 $\{P_1$: an **integer-type object** indicating the number of play points in the first test case},
 $\{INFO_1$: an **array-type object** corresponding to the information of all cards in the first test case},
 $\{C_2$: an **integer-type object** indicating the number of cards in the second test case},
 $\{P_2$: an **integer-type object** indicating the number of play points in the second test case},
 $\{INFO_2$: an **array-type object** corresponding to the information of all cards in the second test case},
 ...
 $\{C_N$: an **integer-type object** indicating the number of cards in the N-th test case},
 $\{P_N$: an **integer-type object** indicating the number of play points in the N-th test case},
 $\{INFO_N$: an **array-type object** corresponding to the information of all cards in the N-th test case}

P.S. $1 \leq N \leq 10, 0 \leq C \leq 10,000, 0 \leq P \leq 10,000$

The data type of each element in information arrays is string. The format of each information array looks like below:

$[$ "<type₁> <cost₁> <attack₁> <defense₁>",
 "<type₂> <cost₂> <attack₂> <defense₂>",
 ...
 "<type_c> <cost_c> <attack_c> <defense_c>" $]$

output.txt

Your output **DO NOT** encode to binary format. You only need to write the results of all cases in a text file looks like below:

{ Maximun damage }₁
 { Maximun damage }₂
 ...
 { Maximun damage }_N

Example

Input (after decode)

```
2
3 10
follower 1 1 1
follower 2 2 1
follower 4 1 4
5 7
spell 1 1 0
follower 3 2 2
follower 4 3 3
spell 6 4 0
follower 7 5 7
```

Note that there is *no newline character* in any string, and there is a space between each attribute of a card.

Output

```
6
7
```

Note that there is *a newline character* at the end of each line.

Discussion Board

If you have any problem regard this homework, please post your question in our [discussion board](#). You are encouraged to post the question on discussion board as others might have the similar question as you do.