HOMEWORK 10

ALGORITHMS AND DATA STRUCTURES (CH08-320201)

Spring 2018

Prof. Dr. Michael Sedlmair
Computer Science & Electrical Engineering
Jacobs University

Due on Monday, May 07, 2018, 23:55.

All problems need to be implemented in python or C++.

Problem 1: Longest ordered subarray

(4 points)

Consider the array A=($a_1, a_2, ...a_n$). We call a subarray a succession of numbers in A where the position of any value in the subarray in the array is always bigger than the value of the previous one. Use dynamic programming to find a subarray of maximal ordered length of the array A. You can assume there are no duplicate values in the array and that there exists just one optimal solution.

Example
For A=(8,3,6,50,10,8,100,30,60,40,80) the solution is: (3,6,10,30,60,80)
Your program should take an input and spit out an output as follows:

Sample input

```
8 3 6 50 10 8 100 30 60 40 80
```

Sample output

```
3 6 10 30 60 80
```

Problem 2: Sum in triangles

(5+1+1=7 points)

Consider a triangle formed from n lines ($1 < n \leq 100$), each line containing natural numbers between [1,10000].

a) Use dynamic programming to determine the biggest sum of numbers existent between the road from the number in the first line and a number from the last line and print the respective road to the output. Each number in this road is seated to the left or to the right of the other value above it.

b) Analyze the runtime of your solution and compare it to a brute force approach.

c) Explain why a greedy algorithm does not work for this problem.

Example
The values are displayed for example in this manner:

$$
\begin{array}{ccccccccc}
 & & & & 7 & & & & \\
 & & & 3 & & 8 & & & \\
 & & 8 & & 1 & & 0 & & \\
 & 2 & & 7 & & 4 & & 4 & \\
4 & & 5 & & 2 & & 6 & & 5
\end{array}
$$

For this example the resulting sum would be 30.
Your program should take an input and spit out an output as follows:

Sample input

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

Sample output

```
30
7 3 8 7 5
```

Problem 3: Scuba Diver *(12 points)*

A scuba diver uses a special equipment for diving. He has a cylinder with two containers: one with oxygen and the other with nitrogen. Depending on the time he wants to stay under water and the depth of diving the scuba diver needs various amount of oxygen and nitrogen. The scuba diver has at his disposal a certain number of cylinders. Each cylinder can be described by its weight and the volume of gas it contains. In order to complete his task the scuba diver needs specific amounts of oxygen and nitrogen. Theoretically, the diver can take as many cylinders as he wants/needs. Use dynamic programming to find the minimal total weight of cylinders he has to take to complete the task and which those cylinders are. In case of several acceptable solutions, printing just one of them is enough.

Example
The scuba diver has at his disposal 5 cylinders described below. Each description consists of: volume of oxygen, volume of nitrogen (both values are given in liters) and weight of the cylinder (given in decagrams):

```
3 36 120
10 25 129
5 50 250
1 45 130
4 20 119
```

If the scuba diver needs 5 liters of oxygen and 60 liters of nitrogen then he has to take two cylinders of total weight 249 (for example the first and the second ones or the fourth and the fifth ones).

Input
The number of test cases c is in the first line of input, then c test cases follow separated by an empty line.

In the first line of a test case there are two integers t, a separated by a single space, $1 \leq t \leq 21$ and $1 \leq a \leq 79$. They denote volumes of oxygen and nitrogen respectively, needed to complete the task. The second line contains one integer n, $1 \leq n \leq 1000$, which is the number of accessible cylinders. The following n lines contain descriptions of cylinders; i-th line contains three integers t_i, a_i, w_i separated by single spaces, $(1 \leq t_i \leq 21,\ 1 \leq a_i \leq 79, 1 \leq w_i \leq 800)$. These are respectively: volume of oxygen and nitrogen in the i-th cylinder and the weight of this cylinder.

Output
On the first line will be printed the total weight w_t and on the next line separated by spaces the index of the cylinders in order.

Sample input

```
1
5 60
5
3 36 120
10 25 129
5 50 250
1 45 130
4 20 119
```

Sample output

```
249
1 2
```

Remarks

- Solutions have to be handed in via Moodle by the due date. For late submissions you need to get in contact with the TAs directly. You need to upload one zip-file that contains a PDF-file for the theoretical parts and source files (no executables or object files) for the programming assignments. The source files need to include a makefile. Programming assignments need to be handed in as **C++, or Python** code. **Please write your own code.** It is ok to take snippets from online resources, but they need to be clearly marked.
- Exercises marked with a * are bonus problems. These count towards your number of points for this homework. The maximum number of official points can not be exceeded.