Data Structure and Algorithm II

Homework #3

Due: 9pm, Thursday, April 14, 2011

=== Homework submission instructions ===

- Submit the answers for writing problems (including your programming report) through the CEIBA system (electronic copy) or to the TA in R204 (hard copy). Please write down your name and school ID in the header of your documents. You also need to submit your programming assignment (problem 1) to the Judgegirl System(`http://katrina.csie.ntu.edu.tw/judgegirl/`).

- Each student may only choose to submit the homework in one way; either all as hard copies or all through CEIBA except the programming assignment. If you submit your homework partially in one way and partially in the other way, you might only get the score of the part submitted as hard copies or the part submitted through CEIBA (the part that the TA chooses).

- If you choose to submit the answers of the writing problems through CEIBA, please combine the answers of all writing problems into only one file in the doc/docx or pdf format, with the file name in the format of "hw3_[student ID].{pdf,docx,doc}" (e.g. "hw3_b98902010.pdf"); otherwise, you might only get the score of one of the files (the one that the TA chooses).

- For each problem, please list your references (they can be the names of the classmates you discussed the problem with, the URL of the information you found on the Internet, or the names of the books you read). The TA can deduct up to 100% of the score assigned to the problems where you don't list your references.

Problem 1. (30%) **Box Stacking.**

You are given a set of n types of rectangular 3-D boxes, where the i-th box has height h_i, width w_i and depth d_i (all real numbers). You want to create a stack of boxes which is as tall as possible, but you can only stack a box on top of another box if the dimensions

of the 2-D base of the lower box are each strictly larger than those of the 2-D base of the higher box. Of course, you can rotate a box so that any side functions as its base. You are also allowed to use multiple instances of the same type of boxes.

Input

The first line has n, which is the number of different types of boxes. $1 \leq n \leq 1000$. The following n lines list the dimensions of different types of boxes. Each of the dimensions will be a real number smaller than 1000 and larger than 0.

Output

The first line shows H, which is the height of the tallest possible stack of boxes using the types of boxes specified in the input. Assume that there are m boxes in the tallest possible stack, the following m lines should list the base dimension of the boxes in the tallest possible stack of boxes, from the top to the bottom of the stack. For each line, we list the width followed by the depth of the base of the box.

Write a program to solve this problem using dynamic programming. Please also submit a report in which you give a clear description of your algorithm.

Problem 2. Read the paragraph "Greedy versus dynamic programming" on p.425-427, then solve the following problems on the textbook:

1. (4%) 16.2-1 on p.427
2. (10%) 16.2-2 on p.427
3. (4%) 16.2-3 on p.427
4. (4%) Which problem that we have talked about in the class reminds you of the 0-1 knapsack problem? Please explain why by giving a problem reduction (problem transformation).

Problem 3. Consider the multistage graph $G = (V, E)$ as shown in Figure 1. Each edge in G is assigned with a nonnegative weight.

1. (12%) Find a shortest and a longest path between S and T using dynamic programming.

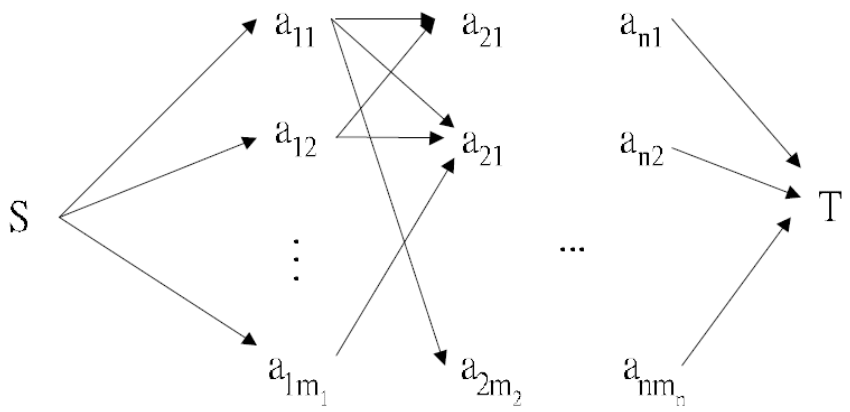

Figure 1: A multigraph

2. (4%) Does your algorithm work if the weights are allowed to be negative? Why?

Problem 4. (12%) Solve Problem 15.5-1 on p.403 of the textbook.

Problem 5. (20%) Solve Problem 15-3 on p.405 of the textbook.