Data Structure and Algorithm II

Homework #2

Due: 13pm, Monday, October 31, 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 R432 (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 "hw2_[student ID].{pdf,docx,doc}" (e.g. "hw2_b99902010.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%) It is always very nice to have little brothers or sisters. You can tease them, lock them in the bathroom or put red hot chili in their sandwiches. But there is also a time when all meanness comes back!

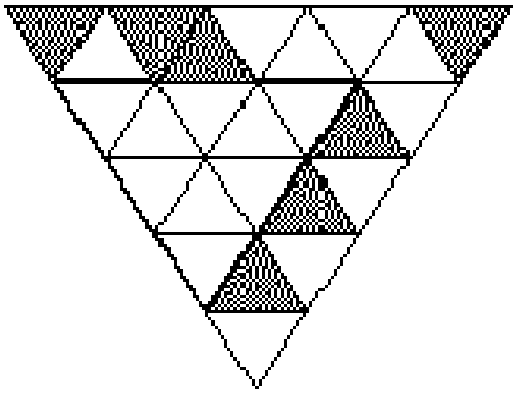

Figure 1: Triangles

As you know, in one month it is Christmas and this year you are honored to make the big star that will be stuck on the top of the Christmas tree. But when you get the triangle-patterned silver paper you realize that there are many holes in it. Your little sister has already cut out smaller triangles for the normal Christmas stars. Your only chance is to find an algorithm that tells you for each piece of silver paper the size of the largest remaining triangle.

Given a triangle structure with white and black fields inside you must find the largest triangle area of white fields, as shown in Figure 1.

Input:

The input file contains several triangle descriptions. The first line of each description contains an integer n ($1 \leq n \leq 100$), which gives the height of the triangle. The next n lines contain characters of the set {space, #, -} representing the rows of the triangle, where '#' is a black and '-' a white field. The spaces are used only to keep the triangle shape in the input by padding at the left end of the lines. (Compare with the sample input. The first test case corresponds to the figure.)

For each triangle, the number of the characters '#' and '-' per line is odd and decreases from 2n - 1 down to 1. The input is terminated by a description starting with n = 0.

There are at most 1000 triangles in each test case. Time limit for this problem is 3 seconds.

Output:

For each triangle in the input, first output the number of the triangle, as shown in the sample output. Then print the line "The largest triangle area is a.", where a is the number of fields inside the largest triangle that consists only of white fields. Note that the largest triangle can have its point at the top, as in the second case of the sample input. Output a blank line after each test case.

Sample Input:

```
5
#-##----#
 -----#-
  ---#-
   -#-
    -
4
#-#-#--
 #---#
  ##-
   -
0
```

Sample Output:

```
Triangle #1
The largest triangle area is 9.

Triangle #2
The largest triangle area is 4.
```

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. (20 points for 10 test cases and 10 points for the report.)

Problem 2. Solve the following problems on the textbook:

1. (5%) 15.2-1 on p.378
2. (5%) 15.2-3 on p.378
3. (10%) 15.4-5 on p.397

Problem 3. (15%) In the class, we have talked about how to use the divide-and-conquer technique to solve the maximum subarray problem. In this problem, we ask you to use dynamic programming to solve this problem. Describe your algorithm and show that the running time of your algorithm can be better than $\Theta(n \log n)$.
Here we re-state the maximum subarray problem. Given an array A, find the nonempty, contiguous subarray of A whose values have the largest sum. The output should include the start and end indices and the sum of the maximum subarray of A.

Problem 4. (15%) Solve problem 15-2 on p.405 of the textbook.

Problem 5. (20%) Solve problem 15-5 a. on p.406-407 of the textbook.

Problem 6. (+5% bonus) We are about a month into the semester now. How do you feel about this course? How much time did you spend on the homework 1 and 2? Please also give some constructive suggestions to the course about the homework, the lectures, or another other related things. Feel free to say ANYTHING about the course. :) (You will get all 5-point bonus if your suggestions or comments are somehow constructive.)