Data Structure Assignment 5

Programming homework 1

Monk And Champions League

Monk's favorite game is Football and his favorite club is "Manchester United". Manchester United has qualified for the Champions League Final which is to be held at the Wembley Stadium in London. So, he decided to go there and watch his favorite team play. After reaching the stadium, he saw that many people have lined up for the match tickets. He knows that there are **M** rows in the stadium with different seating capacities. They may or may not be equal. The price of the ticket depends on the row. If the row has **K**(always greater than 0) vacant seats, then the price of the ticket will be **K** pounds(units of British Currency). Now, every football fan standing in the line will get a ticket one by one.

Given the seating capacities of different rows, find the maximum possible pounds that the club will gain with the help of the ticket sales.

Input:

First line of input contains number of test cases **T** ≤ 10,the second line consists of **M** and **N**. **M** denotes the number of seating rows in the stadium and **N** denotes the number of football fans waiting in the line to get a ticket for the match.

Next line consists of **M** space separated integers **X[1],X[2],X[3].... X[M]** where **X[i]** denotes the number of empty seats initially in the i^{th} row.

Output:

Print in a single line the maximum pounds the club will gain for each case.

Constraints:

1<= **T** ≤ 10
1 <= **M** <= 1000000
1 <= **N** <= 1000000
1 <= **X[i]** <= 1000000
Sum of **X[i]** for all **1** <= **i** <= **M** will always be greater than **N**.

Note:
Write your own implementation of max-heap.
Zero points will be given if you do not use max-heap, and make sure your array is created by dynamic memory allocation or it will stack overflow in some case

Please name your submitted file as "max_heap.c/max_heap.cpp" for this homework

Input.

3
3 4
1 2 4
4 5
4 3 4 7
5 7
2 4 3 1 2

Output:

11
26
18

Explanation

For first sample test case, number of rows is 3 and the 4 people waiting in line to get a ticket.
Since the maximum cost of ticket initially is 4 pounds, therefore the first person in line will buy a ticket for the 3rd row.
The person standing in line will again choose the 3rd row as it has the maximum number of seats, which will cost him 3 pounds.
The next person will have 2 choices, he can either choose the 2nd row or the 3rd row which will cost him 2 pounds.
Similarly, the last person will choose the row will 2 seats remaining, which will cost him 2 pounds.
Total cost = 4+3+2+2 = 11 pounds.

Programming Homework 2

Height Union

Write a function `collapsingFind` (Program 5.21) that incorporates the collapsing rule, and a function `heightUnion` that uses the height rule for union oprations instead of the weighting rule. This rule is defined below:

Definition [Height Rule]: If the height of tree i is less than that of tree j, then make j the parent of i, otherwise make i the parent of j. □

In your readme file, compare `weightedUnion` and `heightUnion` to determine which one produces better results when used in conjunction with function `collapsingFind`.

Input:

First line of input indicates the number of trees T $(1 \leq \mathrm{T} \leq 100)$.

For each tree t, the first line of input consists of a single integer $\mathrm{N_t}$ $(1 \leq \mathrm{N_t} \leq 100)$ which indicates the number of nodes in tree t. Each of the following $\mathrm{N_t}$ lines consists of two integers p and q, which is the node id and its parent id respectively. If p is root, q will be a negative number indicating the tree height.

Then, there will be several rows of input consisted of the following operations:

`UNION i j` use `heightUnion` to perform union on i and j, where both i and j are the roots of their respective trees.

`FIND p` use `collapsingFind` to find the root of node p.

`STOP` indicates the end of inputs.

Output:

For each `FIND` operation, print in a single line the number of moves taken (number of links traversed + number of links resetted). Print all outputs at once after receiving `STOP`.

Input

```
3              (there are 3 trees)
3              (the 1st tree has 3 nodes)
0 -2           (root of 1st tree = 0, tree height = 2)
1 0
2 0
3              (the 2nd tree has 3 nodes)
3 -3           (root of 2nd tree = 3, tree height = 3)
4 3
5 4
1              (the 3rd tree has 1 node)
6 -1           (root of 3rd tree = 6, tree height = 1)
UNION 0 3
FIND 5
FIND 1
FIND 1
STOP
```

Output

```
3              (2 links traversed + 1 link reset)
3              (2 links traversed + 1 link reset)
1              (1 link traversed)
```

Note:

The red text in example is just for clarification purpose and should not be included in the actual input/output.

You may copy the function `collapsingFind` from the textbook, but change it so that it does not perform an extra link reset as stated in Example 5.4.

Please name your submitted file as "height_union.c/height_union.cpp" for this homework.

General information:

- Deadline: **2019/12/15 23:55.**
- Submit your programming assignment to Moodle system.
- Submitted file format: student-ID_Name.zip, e.g. F12345678_王曉明.zip
- Your submitted file must contain Source Code & Readme file (Program description)
- Late homework will not be accepted
- There is a “zero tolerance” for plagiarism. You will receive a score of zero if you get caught plagiarizing.

資料結構課程規定

1. 程式執行環境: Windows、Linux。
2. 程式語言: C/C++
3. 程式作業只需提供 source code 和 readme 的說明文件。Source code 只接受.cpp 和.c 檔,其餘檔案類型恕不接受,說明文件請含括您的程式內容的解說,例如,程式執 行流程,程式架構,如何設計功能等,請不要複製題目或複製程式碼註解貼上。
4. 紙本作業請列出推論過程,僅列出答案而未列出過程者,不給分。繳交紙本作 業時,記得寫名字以及學號,請務必記得用 A4 紙張作答,若超過一張,請自行裝 訂起來再交給助教,否則將斟酌扣分。
5. 紙本作業與程式作業請勿抄襲,如有發現一律 0 分計算。
6. 程式作業上傳至 moodle 各章節底下的繳交區。
7. 程式作業與手寫作業皆不接受遲交與補交。程式作業在公布之後的兩個禮拜內 將會開放上傳繳交,請同學們盡早完成作業,避免在最後期限內的一、兩個小時上 傳 moodle 導致發生問題。
8. 每個程式的程式分數佔 80%,說明文件佔 20%,若並未完成作業題目所有要求、 所交程式碼無法執行或執行結果錯誤,將會依照題目要求和執行結果的完程度評分,其餘的評分項目由當次作業批改的助教來決定。
9. 每次作業壓縮檔(例如 zip, 7z 檔)名稱必須以學號命名,若未註明一律扣該次成績 20 分。

本課程的 TA 時段如下:
星期二 (Tues.) p.m. 3:00 - 5:00
星期四 (Thu.) p.m. 6:00 - 8:00

若是同學有資料結構相關問題可至高速網路實驗室(資訊系新大樓 5F,65503 室)詢 問,為免同學撲空,如同學無法在助教時間前來,煩請事先與助教預約時間。如有 任何問題請寫信給助教。

助教 何允中 e-mail: p76071323@mail.ncku.edu.tw

潘文傑 e-mail: p76081433@mail.ncku.edu.tw

蘇奕瑋 e-mail: p76081108@mail.ncku.edu.tw

張傑閔 e-mail: p76084499@mail.ncku.edu.tw

蘇雨菲 e-mail: p76075068@gs.ncku.edu.tw

Course Provisions

1. Program execution environment: Windows, Linux.
2. Programming language : C/C++ (Languages other than C/C++ are not accepted).
3. Submitted programming homework must include source code in .cpp or .c type, and readme document. You are required to address the **program architecture**, **program functions and how you design your program** in readme file. Do not just write the pseudo code or even just copy and paste your code!
4. You won't get any point for paper homework if you only write the answers without addressing your process and reasons. **Please do your work on A4 papers. If there is more than one page, please staple them together, and write your student id & name on each page**. Points will be deducted otherwise.
5. **There is a "zero tolerance" for plagiarism. You will receive a score of zero if you get caught plagiarizing.**
6. Please submit your programming homework to moodle.
7. Late homework is not accepted.
8. Programming homework grade is divided into two parts: 80% for the code and 20% for the readme file. **Partial points will still be awarded if the output results of your program are partly correct**. The remaining grading standards are decided by the TAs.
9. **Please name the filename of your submitted compressed file (e.g. zip, 7z) after your** student ID number. 20 points will be deducted otherwise.

TA time of the course:
Tues. 15:00 - 17:00
Thu. 18:00 – 20:00

If you have any questions, please come to our lab at TA time (CSIE Bldg. Room 65503). If you are not available to come at TA time, please make another appointment with the TAs. You can also mail us about your questions.

TA 何允中 e-mail: p76071323@mail.ncku.edu.tw

潘文傑 e-mail: p76081433@mail.ncku.edu.tw

蘇奕瑋 e-mail: p76081108@mail.ncku.edu.tw

張傑閔 e-mail: p76084499@mail.ncku.edu.tw

蘇雨菲 e-mail: p76075068@gs.ncku.edu.tw