COMP3600/6466 — Algorithms
Convenor: Hanna Kurniawati
Lecturer: Hanna Kurniawati & Pascal Bercher
E-mail: comp_3600_6466@anu.edu.au

Assignment 3
Due: Tuesday, 2 November 2021 23:59 Canberra Time
Grace Period Ends: Wednesday, 3 November 2021 13:00 Canberra Time
Late Penalty: 100%

IMPORTANT NOTES (PLEASE READ BEFORE WORKING ON THE SOLUTION):

- Please read the entire of this question sheet before starting to work on the solution.
- Your assignment should consist of a single .pdf, named A3-[studentID].pdf and a single .cpp/.java file, named A3-[studentID].cpp or A3-[studentID].java. You should zip these two files together into a single file, named the .zip file A3-[studentID].zip and submit via Wattle before the due date.
- We provide 13 hours grace period. This means, there will be no penalty if you submit before the grace period ends. However, we will NOT accept assignment submission beyond this time (i.e., late penalty after the grace period ends is 100%).
  - You can update your assignment until the grace period ends.
- Assignment marking:
  - The total mark you can get in this assignment is 100 points.
  - This assignment will contribute 20% to your overall mark.
  - In each question, you need to provide a convincing argument (including mathematical derivation) that supports your answer. The argument is worth 80% of the marks, while the solution alone is 20%.
- Discussion with your colleagues are allowed and encouraged. However, you need to work on the assignment on your own AND write the names you discussed this assignment with.

Errata:

- 16 Oct: Part A Q1, sorted order should be sorted array.
- 22 Oct: Clarification on Part B Q4.
- 29 Oct: Clarification on Part B Q4: $P(i, j)$ for $i \neq j$; $i, j \in [1, n-1]$ is supposed to be $P(i, j)$ for $i \neq j$; $i, j \in [1, n]$. But, if you have interpret this as the available direct and in-direct flights are only between city-i and city-j, where $i, j \in [1, n-1]$, we will mark it as correct too.
- 29 Oct: Clarification on Part B Q6: Project is the same as proposal. In other words, you can replace "to fund exactly one project" with "to fund exactly one proposal". The fact that project is the same as proposal is also revealed in the example of input–output we have provided.

[30 pts] Part A

1. [15 pt] Mr Sort were provided with the source code that performs simple uniform hashing with chaining. He would like to modify the chaining component, such that each list is kept in a sorted ~~order~~ **array** using merge sort. What will the effect of this modification be to:
   (a) [5pt] The time complexity of searching for a key
   (b) [5pt] The time complexity of insertion
   (c) [5pt] The time complexity of deletion
2. [8 pt] Mr B has created a complete binary tree, denoted as T and coloured all nodes to be black. He claimed that T is a Red-Black tree. Is he correct? Please justify your answer.
3. [7 pt] In an AVL tree, where is the node with the minimum key located? And where is the node with the maximum key located?

[40 pts] Part B

4. [20 pt] Due to travel restrictions, the global travel cost has sky-rocketed. However, Ms Niche found that if a passenger is willing to change flight multiple times, they can generally get to their destination with a much cheaper flight ticket. Due to her extensive network with airline executives, Ms Niche can purchase any airline tickets free of commission. Therefore, she can charge a flat fee –that is, the fee remains the same regardless of the number of hops to go from one city to a destination city. This flat fee means the total price for travelling from city-1 to city-n via sequence $s = \langle s_1, s_2, \ldots, s_m \rangle$ where s_1 = city-1 and s_m = city-n (denoted as ~~$P(1,n)$~~ $\mathbf{TP}_s(1,n)$), can be computed as ~~$P(1,n)$~~ $\mathbf{TP}_s(1,n) = \sum_{i=1}^{|s|} P(s_i, s_{i+1}) + C$, where C is the flat fee. Ms Niche knows that the prices of all direct flights (i.e., $P(i,j)$ for $i \neq j,\ i,j \in [1,n]$) changes quarterly, and so she would like to develop a program such that she can answer customers' queries on finding the cheapest sequence of cities to travel from city-i to city-n. Please help Ms Niche by answering the following questions:

   (a) [5 pt] Can Ms Niche use Dynamic Programming to solve the above problem? If she can, please show that the two requirements for a problem to be solvable by Dynamic Programming is satisfied. Otherwise, please explain which requirements are being violated.

   (b) [10 pt] Please provide an algorithm that Ms Niche can use to solve her problem.

   (c) [5 pt] Please provide the time complexity of the algorithm you proposed in 4b.

5. [20 pt] The Arrebnac Mall is preparing for their sale of the year. To attract shoppers to spend more, the sale organisers plan to give shoppers a set of round shopping boxes (illustrated in Figure 1), arranged in a nested fashion –that is, these boxes will be arranged as a sequence of boxes with strictly decreasing radius and height, i.e., $r_i > r_{i+1}$ and $h_i > h_{i+1}$ (r_i and h_i are the radius and height of the box at index-i), such that the box at index-$(i+1)$ can be placed inside the box at index-i, where $i \in [1, k-1]$ and k is the number of boxes in the sequence.

   Due to the difficulty in logistics during the pandemic, the sale organisers plan to use the round boxes available in the mall's warehouse. The organisers have taken inventory and found that there's a total of n different sizes (radius and height) of round boxes in the mall's warehouse. Their goal is to choose the sequence of sizes that allows them to build the largest number of boxes in the nested arrangement, which means the sequence will be the longest and satisfy the strictly decreasing order for both radius and height.

   

   Figure 1: Illustration of round boxes

   The IT Department of Arrebnac Mall has asked you to develop an algorithm to find the above sequence and compute the maximum number of boxes in the sequence. For this purpose, please answer the questions below. You can assume that for each size, an extremely large number of boxes with such a type are available, one can assume there's infinitely many boxes of each size.

   (a) [5 pt] Given a list of n box size $[(r_1, h_1), (r_2, h_2), \cdots, (r_n, h_n)]$, please show that the above problem can be solved using Dynamic Programming.
   Note: The above problem can be formally framed as one of finding the longest strictly decreasing sequence of box sizes, such that $r_i > r_{i+1}$ and $h_i > h_{i+1}$ for all $i \in [1, k-1]$, where $k \leq n$ is the length of the sequence.

   (b) [10 pt] Please provide a dynamic programming algorithm that can solve the above problem. You can use either top-down or bottom up dynamic programming.

   (c) [5 pt] Please provide the time complexity of the algorithm you proposed in 5c.

[30 pts] Part C

6. In view of the recent pandemic, the Senate of Planet Htrae has created a new Strategic Health and Medical Research Consortium (SHMRC). This consortium will be provided with $\$F$ millions ($500 \leq F \leq 5,000$) funding per year for strategic medical research to ensure the planet will be ready to fight lethal new diseases. To this end, the consortium director, Mr Muitrosnoc, has compiled a list of n ($n \leq 250$) new and game-changing

approaches to develop immunity against potentially lethal diseases. For each approach-i ($i \in [1, n]$), a panel of experts reviews the proposals and provides Mr Muitrosnoc with k_i ($k_i \in [1, 100]$) proposals per year, and each proposal $j \in [1, k_i]$ requests $\$m_{ij}$ millions ($1 \leq m_{ij} \leq 3 \times \frac{F}{n}$). Mr Muitrosnoc plans to fund exactly one ~~project~~ proposal, so as to encourage collaborations among researchers working on similar approaches. Moreover, he aims to select the proposals, such that the total funding distributed is as high as possible but still within the given budget of $\$F$ millions. The question is how to find a set of such proposals?

A trusted friend of Mr Muitrosnoc informed him that the above problem can be efficiently solved using dynamic programming, though without details. Please help Mr Muitrosnoc by answering the following questions.

(a) [10 pts] Please provide the pseudo-code for bottom-up dynamic programming to help Mr Muitrosnoc.

(b) [5] Please derive the asymptotic time complexity of your bottom-up dynamic programming solution.

(c) [10 pts] Please implement the bottom-up dynamic programming algorithm you have developed above.

Input to the Program is the name of a text (.txt) file. The text file contains $n + 1$ lines, where n is the number of new and game-changing approaches to develop immunity against potentially lethal diseases. The first line consists of two numbers, separated by a single white space. The first number is F and the second number is n. Line-$(i+1)$ for $i \in [1, n]$ consists of $k_i + 1$ numbers, separated by a single white space. The first number represents the number proposals for approach-i, while the next k_i numbers represent the funding m_{ij} requested by the j^{th} proposal in approach-i. You can assume that each number in the input file is a positive integer.

Example: The input to the program is input1.txt, where the content of input1.txt is as follows:
750 5
2 200 300
1 100
3 150 50 100
3 50 80 120
2 400 300

The first line of the above input file means that the available total funding is \$750M and there's 5 types of new and game-changing approaches that will be funded. The subsequent lines provide information about the proposals that have been put forward for each approach. For instance, line-2 implies that for approach-1, 2 proposals have been put forward, with the first proposal requesting \$200M and the second proposal requesting \$300M, while line-3 implies that for approach-2, only 1 proposal, requesting \$100M, has been put forward.

Output of the Program is two lines in standard output. The first line contains $n+1$ numbers separated by a single white space and ended with a line break, where the first number is the total funding distributed and the subsequent numbers are the indices of the selected proposal for each approach in order, i.e., the $(i+1)^{th}$ number is the index of the selected proposal for approach-i. The second line contains n numbers separated by a single white space and ended with a line break, where the i^{th} number is the funding requested by the proposal selected for approach-i.
If more than one solution exist, any set of them will be acceptable.
If the problem has no solution, the output should be "No Solution".
All numbers in the output are in millions.

Example of the output for the input1.txt (input example above):
750 1 1 3 1 2
200 100 100 50 300

Program Marking: If your program compiles and runs, you will get 2 points. We will run your program on 4 test cases. For each test case, your dynamic programming algorithm (including data structure construction) will be given a total of $(0.0001 \times F \times \sum_{i=1}^{n} k_i)$ ms CPU time to find a solution. You can assume your program will have access to at most 12GB RAM. It will be run as a single thread process on a computer with Intel i7 3.4GHz processor. For each test case that your program solves correctly within the given time and memory limit, you will get 2 points.
Note: The test cases for marking will be different from the example test cases.

Tips: Additional test cases are available in A3-prgInfo. However, these are just examples. You should NOT extract specifications from these test cases. Specifications should follow the description in this question sheet.

(d) [5 pts] Please provide experimental analysis of the time complexity of your algorithm and comparison against the theoretical analysis (6b). You will need to have sufficient data to fit a function well and will need to generate your own test cases for this purpose.

oOo That's all for the questions folks oOo