CSC311: The Design and Analysis of Algorithms

Computer Science Department — First Semester 1443

King Saud University — Project

---

*This project is divided into two parts, and carries **15%** of your grade, you may work in groups of two or three.*

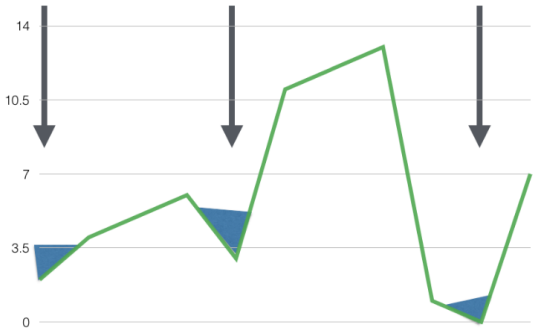
Due:

- **Phase 1:** Thursday, Oct. 7, 2021, at 11:00 P.M.
- **Phase 2:** Thursday, Nov. 11, 2021, at 11:00 P.M.

Phase 1

Suppose that you are interested in collecting rainwater to augment a towns water supply. You know that the water accumulates in valleys (a low altitude point between two higher altitude points), since rain rolls downhill after falling on high points. You want to find the location of one of these valleys, in order to set up a collection point. Assume that the altitude (elevation) is given as a 1D array of numbers at each location (index). For example, here is a possible terrain A=[2, 4, 5, 6, 3, 11, 12, 13, 1, 0, 7], The valleys are A[0], A[4], and A[9]

You are given an array of n elements representing the altitudes of points along the terrain, all



of which are distinct. Design an $O(\log n)$-time algorithm that finds the location of a valley.

Phase 2

Ms. Abeer is arranging for a professional training program. The hierarchical structure of the university forms a tree rooted at the universitys chief executive officer. The employees are ranked by the HR office based on their evaluation, which is a real number. In order to make the best utilization of the program, the chief executive officer does not want both an employee and his or her immediate head to attend. Ms. Abeer is given the tree that describes the structure of the university, using the tree data structure. Each node of the tree holds, in addition to the pointers, the name and evaluation-based ranking of an employee. Describe two algorithms to make up a program registration offer list that maximizes the sum of the evaluation-based ratings of the employees such that an employee does not attend with his/her immediate head, using Brute Force and Dynamic Programming. Analyze the running time of your algorithms.

Deliverables

- The source code (the project folder). Make sure your code is well organized and documented. You may use any language you like, Java, Python, C++ ... etc
- A report that includes the following sections:
  - Cover page
  - Phase 1:
    1. A description of the algorithm (pseudo-code with explanation)
    2. What class of algorithms does your solution fall under?
    3. Proof of correctness (all steps must be clear and justified)
    4. Proof of time complexity $O(\log n)$
    5. Sample run on the provided cases
    6. Source code (in Courier New with font size:8) with comments
    7. Describe challenges faced and how you tackled them
  - Phase 2:
    1. Brute Force:
       (a) A description of the algorithm (pseudo-code with explanation)
       (b) Time and space complexity
       (c) Sample run on the provided cases (see Input Sample)
       (d) Source code (in Courier New with font size:8) with comments
    2. Dynamic Programming:
       (a) A description of the algorithm (pseudo-code with explanation)
       (b) The solution will be for a complete binary tree.
       (c) Time and space complexity
       (d) Sample run on the provided cases (see Input Sample)
       (e) Source code (in Courier New with font size:8) with comments
    3. Describe challenges faced and how you tackled them
    4. Student peer evaluation form (see Team Work Evaluation table below).

Evaluation Rubric

Evaluation rubric is divided into two parts: First part evaluates Team Work and second part addresses the functional requirements. The code in grade assignment is shown to the right.

Code	
Fully satisfied	1
Partially satisfied	0.5
Not satisfied	0

Part 1: Team Work			
Criteria	**Student 1**	**Student 2**	**Student 3**
Work division: Contributed equally to the work			
Peer evaluation: Level of commitments (Interactivity with other team members), and professional behavior towards team & TA			
Project Discussion: Accurate answers, understanding of the presented work, good listeners to questions			
Time management: Attending on time, being ready to start the demo, good time management in discussion and demo.			
Total/2			

Important notes:

- Group size should be 2-3 students.
- No late submissions accepted.
- The group leader is asked to upload everything on LMS as a compressed file (zip) containing (source code and report).
- Discussions will be conducted at the end of the semester.
  - You need to prepare a simple presentation for the discussion, include:
    - The pseudo-code that you will explain.
    - challenges faced and how you tackled them.

Good luck and have fun!

Part 2: Functional Requirements		
	Criteria	**Evaluation**
General	Overall quality of the code implementation (organization, clearness, design,...)	
	Complete report with well organized sections	
	Total/2	
Phase 1	Algorithm description (pseudo-code with explanation)	
	Proof of correctness (all steps must be clear and justified)	
	Proof of time complexity $O(\log n)$	
	Implementation correctness + sample run	
	Total/4	
Phase 2 Brute Force	Algorithm description (pseudo-code with explanation)	
	Time and space complexity	
	Implementation correctness + sample run	
	Total/3	
Phase 2 DP	Algorithm description (pseudo-code with explanation)	
	Time and space complexity	
	Implementation correctness + sample run	
	Total/4	

Input Sample Phase 1

Input: $A = 5, 10, 6, 15$

Output: A[2] with value 6.

The element 6 has two adjacent elements 10 and 15, both are greater than 6.

Input: $A = 10, 8, 15, 2, 23, 12, 67$

Output: A[1] with value 8 **OR** A[3] with value 2 **OR** A[5]with value 12.

The element 8 has two adjacent elements 10 and 15, both are greater than 8, similarly 2 has two adjacent elements 15 and 23, also, 12 has 23 and 67.

Special Cases

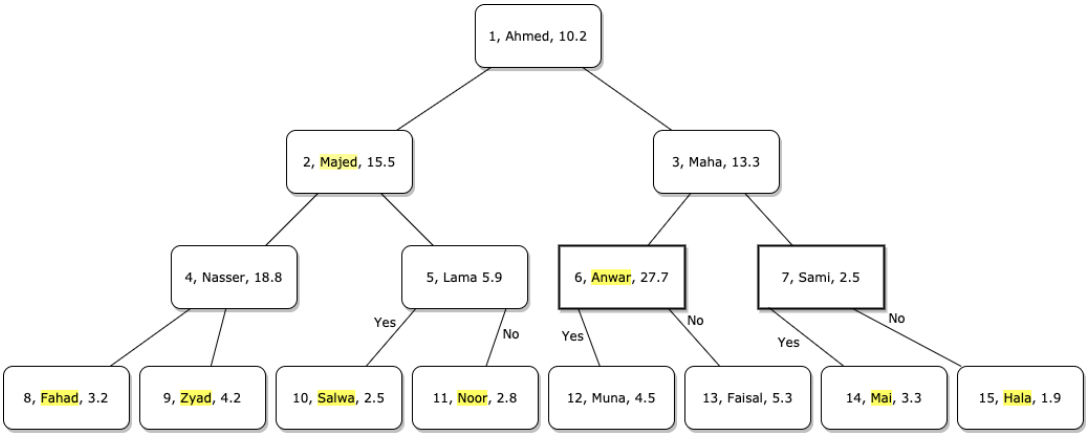
- If the input array A is sorted then:
  - if it in increasing order, the first element is a valley e.g., 1 is the valley in 1,2,3,4,5.
  - if it in decreasing order, the last element is a valley e.g. 1 is the valley in 5,4,3,2,1.
- If all the array elements are the same, every element is a valley

Input Sample Phase 2

Input will be in the form of a text file your program should read. The input has the following format:
ID_of_parent : Name : ID : Evaluation_Score

Example.txt:
0:Ahmed:1:10.2
1:Majed:2:15.5
1:Maha:3:13.3
2:Nasser:4:1.8
2:Lama:5:5.9
3:Anwar:6:12.7
3:Sami:7:2.5
4:Fahad:8:3.2
5:Zyad:9:4.2
7:Salwa:10:2.5

The input is extracted from the following tree:



Output Sample
Program Registration Offer List

ID Name
2 Majed
6 Anwar
8 Fahad
9 Zyad
10 Salwa
11 Noor
14 Mai
15 Hala