CS2223: Algorithms
D-Term, 2017

Assignment 3

Teams: To be done individually

Release date: 04/05/2017

Due date: 04/14/2017 (11:59 PM)

Submission: Electronic submission only

General Instructions

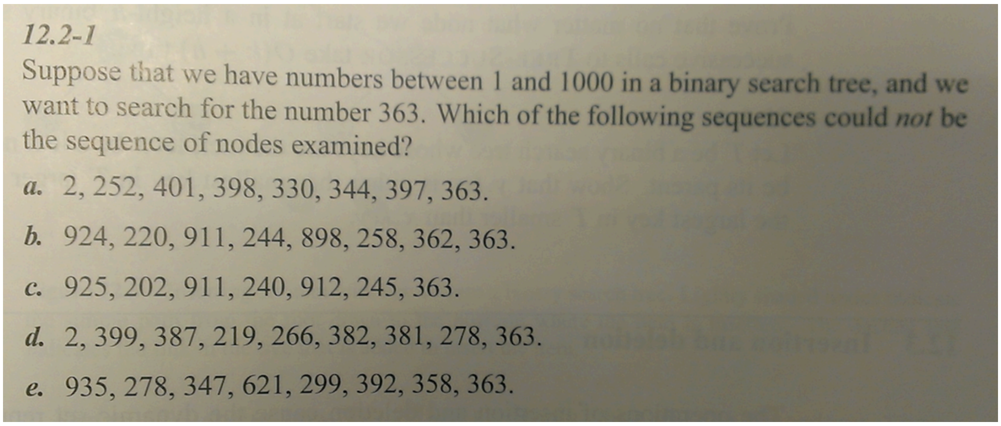
- ***Executable vs. Pseudocode:*** Each question will explicitly state whether the deliverable is pseudocode or an executable program that the TA will run to give you a grade.

- ***Programming Language:*** If a question asking for an executable program, then choose a language from Java, C, C++, or Python. ***You must make it clear in your report:***
  - How to compile your program
  - How to execute it and with which arguments

- ***Submissions:*** The submission of this assignment must be done electronically through blackboard system. All programs plus your report (.doc or .pdf) should be zipped into a single file and that is the file to submit.

Question 1 [30 Points]

1.1) [10 Points] Solve Problem 12.2-1 in your textbook (Page 293). In your report indicate the reason for your choices. //The problem is scanned and included below in case you do not have access to the textbook.



12.2-1
Suppose that we have numbers between 1 and 1000 in a binary search tree, and we want to search for the number 363. Which of the following sequences could *not* be the sequence of nodes examined?

a. 2, 252, 401, 398, 330, 344, 397, 363.

b. 924, 220, 911, 244, 898, 258, 362, 363.

c. 925, 202, 911, 240, 912, 245, 363.

d. 2, 399, 387, 219, 266, 382, 381, 278, 363.

e. 935, 278, 347, 621, 299, 392, 358, 363.

1.2) [10 Points] Assume a binary tree where each node has two fields, a unique ID, and a Value (positive or negative). We need to find the node X where the sum of all values in its subtree (including X) is the maximum.

a) You are asked in this question to sketch a pseudocode for a recursive algorithm that traverses the tree and reports the ID of node X.

b) Analyze your algorithm and state its time complexity.

1.3) [10 Points] Related to binary trees:
Given the following Pre-Order and In-Order traversals of a tree, construct the tree (draw it in your report).

Pre-Order: 10, 3, 5, 4, 15, 7, 8, 2, 9, 20
In-Order: 4, 5, 3, 15, 10, 2, 8, 7, 9, 20

//Hint: A binary tree can be constructed given two of its traversals only if one of the provided traversals is In-Order.

Question 2 [20 Points]

2.1) [5 Points] What is optimal Huffman coding for the following characters, given their frequencies as follows:

a: 3, b: 20, c: 100, d: 50, e: 10, f: 20, g: 5, h: 8

Show in your report the Huffman tree and the final encoding of each character. There can be multiple correct trees.

2.2) [15 Points] Write an executable code that implements Huffman Code algorithm given in class (also in Book 16.3). The program has two modes "encoding" (compressing) and "decoding" (decompressing).

Encoding Mode:

- Your program should accept three input parameters, e.g.,

```
> <YourProgramName>  Encode  <input_file_path_to_compress>  <path_of_output_file>
```

- Your code should compress the file and produce the output to the given <path_of_output_file> location

- *For this mode, the TA will check the size of the compressed file to check if it matches the expected compressed size.*

Decoding Mode:

- Your program should accept three input parameters, e.g.,

```
> <YourProgramName>  Decode  <input_file_path_to_decompress>  <path_of_output_file>
```

- Your code should decompress the input file and produce the output to the given <path_of_output_file> location.

- *For this mode, the TA will apply a diff command to compare the original file with the output file (they should match)..*

Note: You can keep the coding table (Huffman Tree) in memory between the encoding and the decoding, or you can write it to disk a place known to your program.

Note: If the Encoding step is not correct, then you will lose 12 Points. If Encoding is correct but decoding is incorrect, then you lose 8.

Question 3 [10 Points]—Problem Solving

Given an array of unsorted positive and negative values of size N. For example [10, 20, -5, -15, 11, -4, -30, 25, …]. We need to find the two indexes i, and j where the sum of the values between these two indexes (inclusive) is the largest.

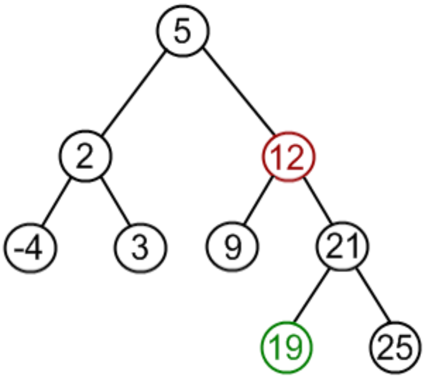
That is, denote by Sum[i,j] the sum of the values between i, j (inclusive), and we want the indexes where there "Sum[]" is the largest.

Write a pseudocode for an algorithm to solve the problem above in O(N). In this question, you only need to sketch the algorithm and analyze it to show it is O(N) time complexity.

Question 4 [10 Points]

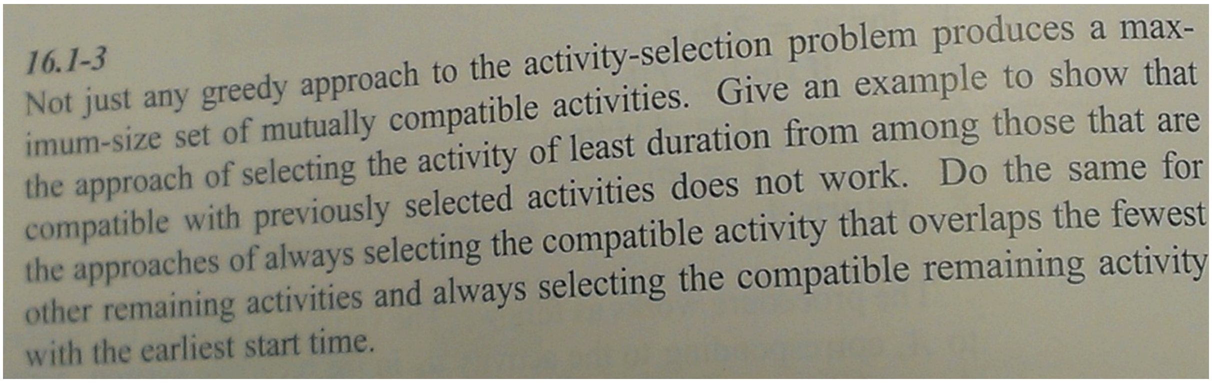
Given the following Binary Search Tree. Show the tree ONCE after the insertions, and ONCE after the deletions. The insertions and deletions are:

- [5 Points] Insertion of 17 followed by the insertion of 4
- [5 Points] Deletion of 5 followed by a deletion of 2



Question 5 [5 Points]

Solve Problem 16.1-3 in your textbook (Page 422). It is included below in case you do not have access to the textbook



16.1-3

Not just any greedy approach to the activity-selection problem produces a maximum-size set of mutually compatible activities. Give an example to show that the approach of selecting the activity of least duration from among those that are compatible with previously selected activities does not work. Do the same for the approaches of always selecting the compatible activity that overlaps the fewest other remaining activities and always selecting the compatible remaining activity with the earliest start time.