SoftEng 284 The University of Auckland

Assignment 2

Due: Monday, August 23rd, by 10pm. *Semester 1, 2021*

There are **four** problems listed below. To get full credit for this assignment you need to complete all of them!

If you are stuck or confused by any of the problems, feel free to post on Piazza! You can also contact the tutors and the lecturer. Note that sometimes we can't answer all your questions (we don't want to be giving away the answers) but we will help you as much as we can.

You must show your working to get full marks You should submit via Canvas a single PDF file containing the answers to the written questions. A scanned handwritten submission is acceptable if and only if it is clearly legible. Hard to read work may not be marked. If typing the assignment, do the best you can with mathematical symbols. For exponents, write something like

```
2^n
```

if using plain text. Use LaTeX if you really want it to look good.

Answers to programming questions must be submitted via the automated marker system: https://www.automarker.cs.auckland.ac.nz/student.php.

Please try to submit your assignments no later than 5 min before the due time. Even though the time is a universal thing your watch and Canvas built-in clock could show the different time. It is quite possible that you might be on time on your watch and late on Canvas. To avoid these kind of situation submit the assignments at least 5 min before the due time.

Please note that late assignments cannot be marked under any circumstances. (With that said: if you find yourself unable to complete this assignment due to illness, injury, or personal/familial misfortune, please email Richard as soon as possible.)

Best of luck, and enjoy the problems!

1. Algorithm analysis (20 marks).

   (a) Consider the following sorting algorithm, sillysort, which when given a list A of size n does the following:

   - Scan the list from left to right, check whether $A[i]$ and $A[i+1]$ is out of order;
   - Swap $A[i]$ and $A[i+1]$ if they are out of order and reset i to the beginning of the list;
   - Increment i by 1 otherwise;
   - Terminate when end of the list is reached.

   Answer the following questions:

   i. Give a pseudo-code for the algorithm sillysort.

   ii. Describe the worst-case input for the algorithm.

   iii. Conduct a worst-case performance analysis of the algorithm and decide whether sillysort performs better, worse or the same with insertion sort asymptotically in the worst case. You need to consider both value comparisons (between items in the input list A) and data movements as elementary operations.

(b) Consider the following sorting algorithm, strangesort, which when given a list A of size n does the following:

- Scan the list from the left to right, comparing each item $A[i]$ with the previous item $A[i-1]$;
- Swap $A[i]$ and $A[i-1]$ if they are out of order and decrement i by 1 (of course, do not decrement i if the start of the list is reach).
- If they are in order, increment i by 1.
- Terminate when end of the list is reached.

Answer the following questions:

i. Give a pseudo-code for the algorithm strangesort.

ii. Analyze the best, worst and average-case performance of strangesort. You need to consider both value comparisons (between items in the input list A) and data movements as elementary operations

2. Recurrence relations (10 marks).

Answer each of the following questions.

(a) The n_{th} **Harmonic number** is defined as $H_n = \sum_{i=1}^{n} i^{-1}$. Solve the following recurrence formula

$$T(n) = \frac{1}{n}(T(0) + T(2) + \cdots + T(n-1)) + 7n$$

with initial condition $T(0) = 0$. You can use the n_{th} Harmonic number as H_n in your answer. Show all working.

(b) Solve the following recurrence formula

$$T(n) = T(\frac{n}{5}) + \log_3 n$$

with initial condition $T(1) = 1$. You may assume n is an exact power of 5.

(c) What is

$$\lim_{n\to\infty} \frac{F(n+1)}{F(n)}$$

where $F(n)$ is the n_{th} number in the Fibonacci sequence defined in class. (Hint: Solving the Fibonacci recurrence formula may help you to determine the answer).

3. Binary trees (5 marks each)

(a) Perform the linear time algorithm to construct a maximum heap on the following array

$$A = [45, 23, 67, 13, 87, 3, 5, 88, 20, 84, 7, 93].$$

Show the heap structure after each iteration.

(b) Consider the following array:

$$A = [4, 33, 6, 90, 33, 32, 31, 91, 25, 89, 50, 33].$$

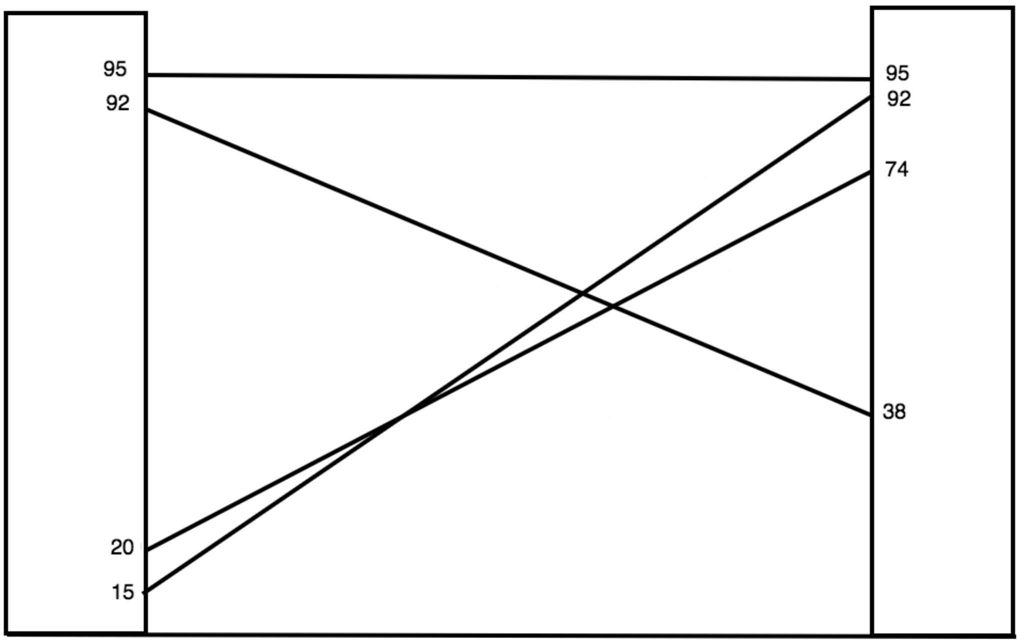
Is A a minimum heap? Justify your answer by briefly explaining the min-heap property. If A is a min-heap, then extract the min value and then rearrange the array with the heapify procedure. If A is not a min-heap, then restore the min-heap property by bubbling up one single value and then extract the min value and heapify.

(c) We have seen in class that the time complexity of many important operations (such as search, update, etc.) associated with BSTs are in order of $\Theta(\phi)$ where ϕ is the average node depth. Given a BST T, suggest an algorithm to compute ϕ efficiently. Analyze the time complexity of your algorithm.

(d) Prove that the height of a complete binary tree with n nodes is exactly $\lceil \lg(n+1) \rceil - 1$ using mathematical induction.

4. (20 marks) **Programming question:**

- **Question:** There are two n-story buildings A and B side by side. Each floor of the building is indexed by an integer in range from 0 to $n-1$. Floor i of A can be connected to floor j of B by a single straight wire. Two wires (i, j) and (i', j') would cross each other in mid air if $i < i'$ and $j > j'$ or $i > i'$ and $j < j'$. See below for an illustration of two 100-story building with four wires between them (note that the picture is not to scale). Given a set of wires in the form of (i, j), your job is to find out how many crosses would this set of wires form. In the example below, there are 3 crosses. You may assume that each floor has at most one wire connected to it.



- **Input:** The first line of the input is a single integer m, denoting the number of scenarios to follow. Each of the next m lines is a separate scenario. The start of each line is an integer n (the height of the two buildings) followed by a list of at most n tuples of the form (i, j) each representing a wire. The beginning and the end of list is marked by '[' and ']' respectively, and each tuple in the list is separated by a single ','.
- **Sample Input:**
  2
  4 [(20,74),(95,95),(15,92),(92,38)]
  3 [(1,2)]
  The first scenario is the same as the one given in the illustration above, and the second scenario represent two 3-story buildings connected by a single wire (so obviously no crosses).
- **Output:** For each line in the input, print a single number that is the number of crosses.
- **Sample Output:**
  3
  0
  This is the corresponding output for the sample input given above.

- **Language:** You can use Java, C,C++, PyPy, Python 3, Go, Rust, F#, Javascript.
- **Note:** There is a limit of 15 submission attempts for this question. This is to prevent people from spamming the automarker (i.e. please don't use the automarker as a debugger, you should test your program carefully before submitting it to the automarker). You should test your program with different inputs, a small sample input and its corresponding output will be on Canvas. Just getting the correct output with the sample is not enough to ensure the correctness of your program.

  Please do not share your code with other students. However, I'm happy if you want to share test cases. You will have to develop your own test cases this time.

  There will be three sets of input for you to submit your program on the automarker. Each has a time limit that your program has to finish within. Something like a brute-force approach probably won't work on the harder input. The small test case is worth 10 marks and the other two test cases are worth 5 marks each so you should at least develop a correct algorithm before you go on to improve it. Number of submissions are counted separately for the different cases (i.e. you can make 15 submissions each).

  The last submission submitted before the assignment deadline will be the one marked.

  Good luck and have fun.
- **Other useful information:** You can assume that input will come from standard input (stdin) in a stream that represents one string per line. Output should be sent to standard output.

  Your can only submit a single program file (containing all nonstandard classes you use, e.g. you can have nested classes if you use Java).

  If your submission was put in the queue before the assignment due time then it will be accepted. All submissions after the assignment due time will not be considered. Exceptions: people who have an extension.

  Start early! Lots of students will be submitting their work closer to the deadline so it may take 30 min before your program will be executed and you will see the results.

  Your output should exactly match the one in the system for the submission to be correct. So be careful with the printing. No extra symbols! It may look the same on the first glance but may have a different end of line character or extra space.

  Please test at the command-line like the following.

  ```
  python3 task1.py < canvas.in > myout1.txt
  diff myout1.txt canvas.out1
  ```

  If you are on a Windows platform you may need to download diff.exe or use alternatives fc or comp.