Homework 8

Data Structures & Algorithms (Term III/2018–19)

built on 2019/07/11 at 16:41:43 ***due:*** *fri july 19 @ 11:59pm*

This assignment contains both a coding portion and a written portion. The aim of this assignment is to provide you with more experience solving algorithmic problems in Java and reasoning about code.

READ THIS BEFORE YOU BEGIN:

- All your work will be handed in as a single zip file. Call this file `a8.zip`. You'll upload this to Canvas before the assignment is due.
- For the written part, you *must* typeset your answers and hand it in as a PDF file called `hw8.pdf`, which will go inside your zip file. No other format will be accepted. To typeset your homework, apart from Microsoft Word, there are LibreOffice and LaTeX, which we recommend. Note that a scan of your handwritten solution will not be accepted.
- Be sure to **disclose your collaborators in the PDF file.**
- For each task, save your work in a file as described in the task description.
- A script will process and grade your submission before any human being looks at it. *Do not use different function/file names.* The script is not as forgiving as we are.
- Use the Internet to help you learn: It's OK to look up syntax or how a function/class/method is used. It's **NOT** OK to look up how to solve a problem or answers to a problem. The goal here is to learn, not to just hand in an answer. If you wish to do that, just give us a link to the solution!
- You are encouraged to work with other students. However, **you must write up the solutions separately on your own and in your own words**. This also means you must not look at or copy someone else's code.
- Finally, the course staff is here to help. We'll steer you toward solutions. Catch us in real-life or online on Canvas discussion.

Collaboration Policy: To facilitate cooperative learning, you are permitted to discuss homework questions with other students provided that the following whiteboard policy is respected. A discussion may take place at the whiteboard (or using scrap paper, etc.), but no one is allowed to take notes or record the discussion or what is written on the board. *The fact that you can recreate the solution from memory is taken as proof that you actually understood it, and you may actually be interviewed about your answers.*

Exercise 1: Mathematical Statements (4 points)

Prove the following statements using your favorite proof method:

(1) Let $G = (V, E)$ be a simple, undirected, weighted graph. A simple graph is one where it has no parallel edges nor self loops.
Proposition A: If the weights of the edges of G are all distinct, then G has a unique minimum spanning tree (MST). *(Hint: contradiction. what can be reused from the light-rule proof?)*

(2) Remember that $[n]$ is the set $\{1, 2, \ldots, n\}$. Let $\pi \colon [n] \to [n]$ be a random permutation on $[n]$ chosen uniformly among the $n!$ permutations. Consider the following code:

```
int minSoFar = Integer.MAX_VALUE;
int numUpdate = 0;
for (int i=1;i<=n;i++) {
    if (π(i) < minSoFar) {
        minSoFar = π(i);
        numUpdate++;
    }
}
```

Prove that at the end of the **for**-loop,

$$\ln(n+1) \le \mathbf{E}[\texttt{numUpdate}] \le 1 + \ln n.$$

Exercise 2: Expectations (6 points)

A *p-biased coin*, $0 \le p \le 1$, turns up heads with probability p and tails with the remaining probability. Answer the following questions in terms of p (show your derivation):

- If you keep on tossing this coin until you see a tails followed directly by a tails (TT), what is the *expected* number of coin tosses, including the final TT?
- If you keep on tossing this coin until you see *three* consecutive tails (TTT), what is the *expected* number of coin tosses, including the final TTT?
- If you keep on tossing this coin until you see THT consecutively, what is the *expected* number of coin tosses, including the final THT?

(Hint: Write probabilistic recurrences)

Exercise 3: HackerRank Problems (6 points)

For this task, save your code in `hackerrank.txt`

There are *six* problems in this set. But you only need to do 3 of them. If you do more, the extra score will be extra credit. You **must** write your solutions in Java (1.8). You will hand them in electronically on the HackerRank website.

Important: You will write down your Hacker ID username in a file called `hackerrank.txt`, which you will submit as part of the assignment. This will be used to match you with your submission on HackerRank.

You can find your problems at

`https://www.hackerrank.com/muic-data-structures-t-318-assignment-8`