Comp 251: Assignment 5

Instructor: Jérôme Waldispühl

Due on April 11th at 11h59

- Your solution must be returned electronically on MyCourse.
- The only format accepted for written answers are PDF or text files (e.g. `.txt` or `.rtf`). PDF files must open on SOCS computers. Any additional files (e.g. images) must be included in the PDF.
- Do not submit compressed files (e.g. zip files). Upload instead a single PDF or text file individually.
- To some extent, collaborations are allowed. These collaborations should not go as far as sharing code or giving away the answer. You must indicate on your assignments the names of the persons with who you collaborated or discussed your assignments (including members of the course staff). If you did not collaborate with anyone, you write "No collaborators" at the beginning of your document. If asked, you should be able to orally explain your solution to a member of the course staff.
- Unless specified, all answers must be justified.
- When applicable, your pseudo-code should be commented and indented.
- The clarity and presentation of your answers is part of the grading. Be neat!
- Violation of all rules above may result in penalties or even absence of grading (Please, refer to the course webpage for a full description of the policy).
- Partial answers will receive credits.
- The course staff will answer questions about the assignment during office hours or in the online forum at `https://osqa.cs.mcgill.ca/`. We urge you to ask your questions as early as possible. We cannot guarantee that questions asked less than 24h before the submission deadline will be answered in time.
- Late submissions will be accepted until April 19^{th} without penalty.

1. (30 points) Suppose we perform a sequence of n operations on a data structure in which the i^{th} operation costs i if i is an exact power of 2, and 1 otherwise. Use aggregate analysis to determine the amortized cost per operation.

2. (30 points) Professor Toole proposes a new divide-and-conquer algorithm for computing minimum spanning trees, which goes as follows. Given a graph $G = (V, E)$, partition the set V of vertices into two sets V_1 and V_2 such that $|V_1|$ and $|V_2|$ differ by at most 1. Let E_1 be the set of edges that are incident only on vertices in V_1, and let E_2 be the set of edges that are incident

only on vertices in V_2. Recursively solve a minimum-spanning-tree problem on each of the two subgraphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. Finally, select the minimum-weight edge in E that crosses the cut (V_1, V_2), and use this edge to unite the resulting two minimum spanning trees into a single spanning tree.
Either argue that the algorithm correctly computes a minimum spanning tree of G, or provide an example for which the algorithm fails.

3. (30 points) When RANDOMIZED-QUICKSORT runs, how many calls are made to the random-number generator RANDOM in the worst case? How about in the best case? Give your answer in terms of Θ-notation with a brief justification.

4. (10 points) End-of-class bonus!