Algorithms and Data Structures 2023: Assignment 3, due October 25

Please read the assignment policies on the course homepage before starting the assignment.

---

1. Reading: (i) Chapter 22 of CLRS. (ii) Section 24 Introduction, 24.1, 24.2 from CLRS.

2. (10 marks) Let $G = (V, E)$ be a connected, undirected graph. Give an $O(m)$-time algorithm to compute a path in G that traverses each edge in E exactly twice: once in each direction.

3. (15 marks) Given a directed graph $G = (V, E)$ with special vertices s and t, we define the following sets. Let X be the set of vertices that *always* lie on the side of s in any minimum cut (e.g., $s \in X$). Let Y be the set of vertices that *always* lie on the side of t in any minimum cut (e.g., $t \in Y$). Let $Z = V \setminus (X \cup Y)$. Give an $O(\text{time for max-flow computation})$-time algorithm to partition V into X, Y, and Z.

4. (5 marks) Given a set S of n items, a function $f : 2^S \to \mathbb{R}$ is said to be *submodular* if, for all sets $A \subseteq B$ and elements $x \notin B$,

$$f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B)$$

   That is, the marginal value of an element to a smaller set, is at least it's marginal value to a larger set.

   Prove that a function f is submodular if and only if it satisfies, for any sets $X, Y \subseteq S$,

$$f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y)\,.$$

5. (5 marks) Let $G = (V, E)$ be a directed graph with nonnegative integral capacity c_e on each edge. Define the *cut function* $f : 2^V \to \mathbb{Z}_+$ as

$$f(S) = \sum_{e=(u,v):u \in S, v \notin S} c_e\,.$$

   Show that the cut function f is submodular.

6. (10 marks) Let $G = (V, E)$ be a directed graph with nonnegative integral capacities on the edges, and let s, t, be two special vertices in the graph. Let $(S, V \setminus S)$ be a minimum s-t cut with vertices u, v in S, **so that there exists a minimum u-v cut $(U, V \setminus U)$ with $t \notin U$.** Then show that there exists a minimum u-v cut $(U', V \setminus U')$ so that $U' \subseteq S$ ~~or $V \setminus U \subseteq S$~~.

   Problems 4, 5 may be useful in solving this.

7. (25 marks) Problem 21-2 from CLRS (the Find-Set procedure is the same as the Find procedure discussed in class for the Union-Find data structure, but you may find it helpful to read Sections 21.3 and 21.4 for this problem).

8. (10 marks) Problem 16-4 a. from CLRS. You don't have to do Part b. of this problem.

9. We are given two red-black trees T_1 and T_2 and an element x, with the guarantee that, for any $x_1 \in T_1$ and $x_2 \in T_2$, $x_1.key < x.key < x_2.key$. Our problem is to implement the procedure RB-Join that forms a single red-black tree from the elements in T_1, T_2, and x. Let n be the total number of nodes in T_1 and T_2.

(i) (5 marks) Given a red-black tree with n' nodes, show that the black-height of the tree can be obtained in time $O(\log n')$. Let $T.bh$ store this information for each red-black tree T.

(ii) (5 marks) Assume that $T_1.bh \geq T_2.bh$. Give an $O(\log n)$ times algorithm that finds a black node y in T_1 with the largest key from among all nodes in T_1 with black-height $T_2.bh$.

(iii) (5 marks) Let T_y be the subtree rooted at y. Describe how $T_y \cup \{x\} \cup T_2$ can replace T_y in $O(1)$ time without destroying the binary search tree property.

(iv) (10 marks) What colour should x be so that the red-black properties 1, 3, 5 (from Section 13.1 of CLRS) are maintained? Describe how to enforce properties 2 and 4 in $O(\log n)$ time.

(v) (5 marks) Complete the description of RB-JOIN, and show the running time.