Sample Problems

Problem 1: TRUE/FALSE Questions with Justifications

For each of the following statements, answer whether it is true or false, and justify your choice in *at most* 3 sentences. Answers exceeding 3 sentences *may not* be read. Answers without justifications will *not* receive full credit. A justfication need *not* be a full proof, and is for demonstration of your understanding only.

1. The partition subroutine of Quick Sort can be modified so that Quick Sort runs in $O(n \log n)$ time in the *worst case*.

2. BFS can be used to decide whether a given *undirected* graph $G = (V, E)$ has a cycle of odd length in time $O(V + E)$.

3. Let T_e be the time for each extract-min operation and T_d that for each decrease-key operation of a give priority queue. Then using this priority queue Dijkstra's algorithm for computing single-source shortest paths runs in time $O(E \cdot T_e + V \cdot T_d)$.

4. To show that a laguange L is **NP-Complete**, it suffices to show that (1) $L \in \mathbf{NP}$, and (2) L reduces to 3SAT in polynomial time.

5. Define the language 3CLIQUE as

$$\text{3CLIQUE} = \{\langle G \rangle : G \text{ has a clique of size } 3\}.$$

   Then 3CLIQUE is **NP-Complete**.

Problem 2: Questions with Short Answers

1. Suppose there are n files $F_1, F_2, \ldots, F_n$ with lengths $l_1, l_2, \ldots, l_n$ respectively, and we would like to concatenate them into a single file $F = F_1 \circ F_2 \circ \cdots \circ F_n$ of length $l = l_1 + l_2 + \cdots + l_n$. The only primitive operation availabe is a binary concatenation operation that concatenates *two* given files F and F' of lengths l and l' respectively, into a single file of length $l + l'$. The cost of this operation on two input files of lengths l and l' is given by a cost function $c(l, l')$. Give an $O(n^3)$ time algorithm that finds an optimal sequence of binary concatenations.

2. Give an efficient algorithm that given a weighted directed graph $G = (V, E)$ with *positive integral* edge weights, and a start vertex s, finds for each vertex $v \in V$ a *shortest path* from s to v *with the smallest number of edges*.