CS3230 2021/22 Semester 2 Final Exam Practice

1. True or False?

   (a) Suppose the amortized cost of an operation `op` over 7 calls to it is 4. It is possible that for 3 out of the 7 calls, the actual cost of `op` is 10.

   (b) You learn about a problem GraphJiraffication that is in NP. You secretly work on the problem for years and finally come up with a polynomial-time algorithm. You have shown $\mathsf{P} = \mathsf{NP}$.

2. Consider the following greedy algorithm for the maximum independent set problem. Find a lowest-degree vertex v and put v in the independent set; remove v and its neighbors from the graph; repeat until no vertices remain.

   Give a counterexample to show that this algorithm does not always yield a correct answer.

3. Suppose you are given an array of n integers $[a_1, \ldots, a_n]$ between 0 and M. Design and analyze an algorithm for dividing the array indices into two sets X and Y such that $\left|\sum_{i\in X} a_i - \sum_{i\in Y} a_i\right|$, the difference of the sum of the integers in each set, is minimized.

   For example, given the array $[2, 3, 2, 7, 9]$, elements at positions $1, 3, 4$ sum up to $2 + 2 + 7 = 11$, while the elements at positions $2, 5$ sum up to $3 + 9 = 12$, yielding a difference of 1.

4. Consider the activity selection problem: You are given a set of n activities with starting times $s_1, \ldots, s_n$ and finishing times $f_1, \ldots f_n$ where each $s_i \leq f_i$. The goal is to find the largest subset of activities which don't *conflict*, meaning that for any pair of activities selected, the finishing time of one of them is not later than the starting time of the other. Consider each of the following greedy strategies. If it works, give a proof of the greedy-choice property. If it doesn't, show a counterexample.

   (a) Choose an activity that ends first, discard all that conflict with it, and recurse.

   (b) If no activities conflict, choose them all. Otherwise, discard an activity that conflicts with the most number of other activities, and recurse.

5. Define the problem PartitionEqual as follows: Given n nonnegative integers, decide whether they can be partitioned into two parts of size $n/2$ each so that both parts have the same sum. Prove that PartitionEqual is NP-complete.

   (You may assume, without proof, that the following problem called Partition is NP-complete: Given nonnegative integers $x_1, \ldots, x_n$, decide whether they can be partitioned into two parts with equal sum.)

6. Given a directed graph G, a *simple path* is a path with no repeated vertices, while a *simple cycle* is a cycle with no repeated vertices. Consider the following two decision problems:

   - LongSimplePath: Given an unweighted **directed** graph G, two vertices u and v, and a positive integer k, decide whether there exists a simple path in G from u to v of length at least k.
   - LongSimpleCycle: Given an unweighted **directed** graph G and a positive integer ℓ, decide whether there exists a simple cycle in G on at least ℓ vertices.

   (a) Describe a polynomial-time reduction from LongSimplePath to LongSimpleCycle. Prove the correctness of your reduction.

   (b) LongSimplePath is NP-hard. From this and part (a), show that there is no known polynomial-time algorithm for LongSimpleCycle.