Spring 2022 – Theory qual part I : Algorithms

1 Seeing is believing.

1.A. Let $G = (V, E)$ be a directed graph with vertex weights $w : V \to \mathbb{Z}$. A set $S \subseteq V$ *sees* the graph if for all $v \in V$ there is a path from some node in S to v. Describe an efficient algorithm that computes the minimum weight set S that can see the graph.

1.B. Consider the version of the above problem where there is a set $X \subseteq V$, and the task is to compute the minimum weight set of vertices S in V, such that S sees all the vertices of X. Prove that this problem is NP-Hard.

2 Long.

2.A. Given a sequence of positive integers $a_1, \ldots, a_n$, describe an efficient algorithm that computes the longest increasing subsequence among subsequences whose sum is 2 mod 5.

2.B. Given a sequence S of n distinct positive integers, describe an efficient algorithm for computing a partition of S into a minimal number of disjoint increasing subsequences. Prove the correctness of your algorithm.

(Hint: Reduce the problem to a graph problem, and solve it.)

3 Cycles. Qual 2016

Let G be a simple regular undirected graph – that is, G has no parallel edges and all the vertices have the same degree. A *cycle cover* is a collection of vertex disjoint cycles that span the vertices of the graph. It is easy to see that a 2-regular graph has a cycle cover, and G itself is the cycle cover.

3.A. Suppose G is 2^k-regular for some $k > 1$. Prove that there is a 2^{k-1}-regular subgraph of G.

3.B. Using the above prove that every 2^k-regular graph has a cycle cover.

3.C. Hard: Prove that every regular even-degree graph has a cycle cover.

4 No such thing as a free lunch. (Qual, Spring 2011)

Suppose that a group of N people (where $N > 50$) are waiting in line to eat at a restaurant. Suppose that the first person has to pay. The second person gets to eat for free if they have the same birthday as the first person. Then everyone else must pay to eat. The third person gets to eat for free if the second person did not get to eat for free *and* they have the same birthday as either the first or the second person. Then everyone else must pay to eat. The "eat free" rule proceeds in the obvious manner – the ith person eats for free if no other person before it eat for free, and their birthday collides with some earlier person.

Determine the probability that the nth person gets to eat for free. Determine the value for n such that their probability of eating for free is larger than anyone else (this last part might require some messy manual calculations – giving a decent approximation, and describing how to get the exact value is enough).

We assume for simplicity that a birthday of a person is being randomly selected uniformly from the 364 days of the year.