CSE 421 Winter 2021
Homework 8

Due: Friday, March 12, 2021, 6:00 pm

Problem 1:
Suppose that you had an algorithm C that on input any (G,k) that correctly answers the Vertex-Cover problem. That is, on input (G,k) for G an undirected graph and k an integer, correctly answers YES if undirected graph G has a vertex cover of size at most k and answers NO if there is no such cover. Show that on any input (G',k'), with a polynomial amount of work and a polynomial number of calls to C you can actually find a vertex cover W of G' of size at most k' if one exists and answer FAIL otherwise.

Problem 2:
At an amusement park there is a dart game that has overlapping regions, each enclosed by its own colored line marked on the wall. Contestants win at the dart game if they can toss the darts at the wall and land at least one dart inside every marked region. A single dart may land inside multiple regions, but they only get k of them.

We consider an abstract version of the dart game, which is a discrete variant with only n places, numbered 1 through n, where a dart could land, and each of the m regions, $R_1,\ldots,R_m$, is just an arbitrary subset of $\{1,\ldots,n\}$. The *Abstract-Dart-Game* problem asks whether there is a set of at most k spots among $\{1,\ldots,n\}$ so that if darts land in all those spots then every region would contain a dart.

Prove that the *Abstract-Dart-Game* problem is NP-complete.

Problem 3:
This problem is a variant of the Interval Scheduling problem that we saw how to solve with a greedy algorithm. In this version of the problem, there is a single resource available for scheduling, but instead of requiring the resource for the whole time between the start time s_i and finish time f_i, the request has scheduled breaks during which time other jobs may use the resource. That is, though job i has an overall start time of s_i and finish time f_i, it may have a break that starts at some time b_{i1} and ends a e_{i1} and, after returning for a while, might have a second break beginning at some b_{i2} and ending at e_{i2}, etc. In general we would have $s_i < b_{i1} < e_{i1} < b_{i2} < e_{i2}... < f_i$ and no bound on the number of breaks.

For example, if we had two requests, one beginning on hour 0 and one beginning on hour 1, each of which repeatedly runs for an hour and then takes an hour break, then the two requests would be compatible with each other and could both be scheduled.

Assume that all start, finish, and break times for all requests are integers and the total time range is between 0 and T. The *Interval-Scheduling-with-Breaks* Problem is to determine. given a collection of descriptions of n requests with breaks and an integer k, whether or not it is possible to schedule at least k of the requests on the single resource.

Prove that the *Interval-Scheduling-with-Breaks* problem is NP-complete.

Problem 4 (Extra credit):
In this problem, as input you are given an $m \times n$ array A of real numbers for which you are promised that the sum of each row and the sum of each column is an integer. Your goal is to find a way of rounding each matrix entry up to the closest integer or down to the closest integer while maintaining all the row and column sums. More precisely, give a polynomial-time algorithm that will produce a new $m \times n$ integer array B such that $B[i, j]$ is $\lceil A[i, j] \rceil$ or $\lfloor A[i, j] \rfloor$, and each row sum or column sum in B is equal to the corresponding sum in A.

For example, if the input is the array on the left, your algorithm could output any of the arrays on the right.

$$\begin{array}{|c|c|c|}\hline 0.4 & 0.1 & 1.5 \\ \hline 0.6 & 1.9 & 0.5 \\ \hline \end{array} \Rightarrow \begin{array}{|c|c|c|}\hline 1 & 0 & 1 \\ \hline 0 & 2 & 1 \\ \hline \end{array} \text{ or } \begin{array}{|c|c|c|}\hline 0 & 0 & 2 \\ \hline 1 & 2 & 0 \\ \hline \end{array} \text{ or } \begin{array}{|c|c|c|}\hline 0 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$