Algorithm Design and Analysis (Fall 2022)
Midterm Exam

Instructions:

1. The exam contains 5 pages. Please write your name and student ID on *every* page.
2. Do not start to read the questions until the examination starts.
3. This is an open book exam.
4. Exam duration: 8:00AM - 9:30AM.
5. For each question, you can continue your solution on the back of the page.

Name: ______________________ **Student ID:** ______________________

Name: ______________________ **Student ID:** ______________________

1. (35 Points) Consider the special case of the Knapsack problem where each item i's value v_i equals to its weight w_i. Suppose $v_i = w_i$ is a (positive) integer power of 2 for each $i = 1, \ldots, n$, and the capacity constraint K is an integer. Consider the following greedy algorithm:

   - sort the items by descending order of weights (or values);
   - for each item i, if putting i into the knapsack does not make the total weight more than K, put i in.

   Does this algorithm always output an optimal solution? If so, prove it. If not, give a counterexample.

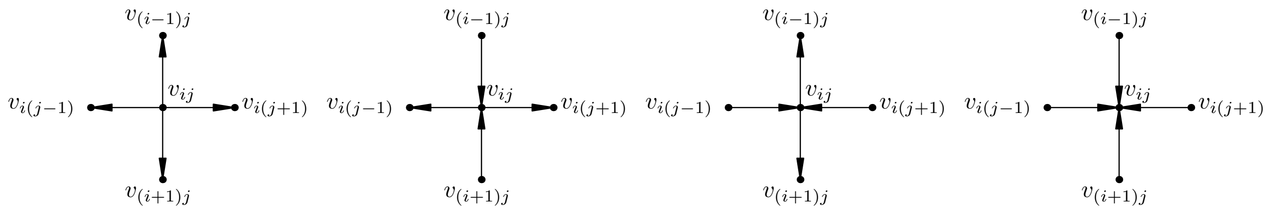
Name: ________________________ **Student ID:** ________________________

2. (a) (15 Points) Let G be a *directed* graph with n vertices labeled $v_1, \ldots, v_n$. The graph contains $n-1$ edges $e_1, \ldots, e_{n-1}$, and it is known that e_i can be either (v_i, v_{i+1}) or (v_{i+1}, v_i). That is, if we ignore the directions of the edges, G is a line. Design an algorithm that finds *one* vertex with out-degree 0. Your algorithm must run in $O(\log n)$ time. Prove the correctness of your algorithm, and analyze its time complexity.

   The adjacency list is used to store the graph, and the indices of the n vertices are stored in an array.

Name: ____________________ **Student ID:** ____________________

(b) (15 Points) Let G be a *directed* graph with n^2 vertices labeled $\{v_{ij}\}_{i=0,1,\ldots,n-1;j=0,1,\ldots,n-1}$. Each vertex v_{ij} has four neighbors $v_{(i-1)j}, v_{(i+1)j}, v_{i(j-1)}, v_{i(j+1)}$, where we adopt the convention that $0-1=n-1$ and $(n-1)+1=0$ (for example, $v_{(n-1)j}$ is a neighbor of v_{0j}). It is known that the directions of the edges between v_{ij} and the "left" and the "right" neighbors of v_{ij} are the same, i.e., either both edges point towards v_{ij} or both edges point outwards v_{ij}, and the directions of the edges between v_{ij} and the "upper" and the "lower" neighbors of v_{ij} are the same. Specifically, the four edges incident to v_{ij} must be in one of the following four configurations.



Design an algorithm that finds *one vertex with out-degree* 0 <u>or</u> *one directed cycle*. Your algorithm must run in $O(\log n)$ time. Prove the correctness of your algorithm, and analyze its time complexity.

The adjacency list is used to store the graph, and the indices of the n^2 vertices are stored in an array.

Name: ______________________ **Student ID:** ______________________

3. (35 Points) Given an *undirected* (unweighted) graph $G = (V, E)$ and a vertex s, design an efficient algorithm to find a cycle with minimum number of edges that contains s. Notice that a cycle cannot contain an edge more than once. In particular, for a neighbor u of s, s-u-s is not a valid cycle. Prove the correctness of your algorithm and analyze its time complexity. Try to design an algorithm with time complexity as lower as you can.