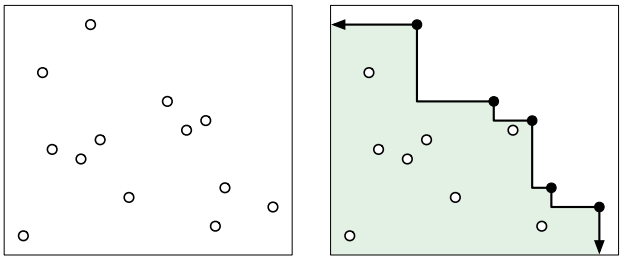
Write your answers in the separate answer booklet.
You may take this question sheet with you when you leave.

1. Let S be an arbitrary set of n points in the plane. A point p in S is ***Pareto-optimal*** if no other point in S is both above and to the right. The ***staircase*** of S is the set of all points in the plane (not just in S) that have at least one point in S both above and to the right. All Pareto-optimal points lie on the boundary of the staircase.



A set of points in the plane and its staircase (shaded), with Pareto-optimal points in black.

   (a) Describe and analyze an algorithm that computes the staircase of S in $O(n \log n)$ time.

   (b) Suppose each point in S is chosen independently and uniformly at random from the unit square $[0,1] \times [0,1]$. What is the ***exact*** expected number of Pareto-optimal points in S?

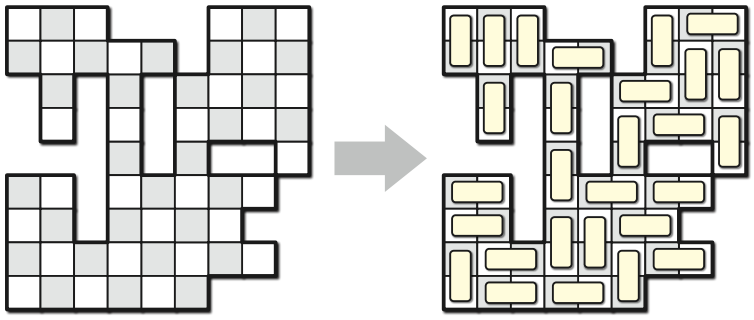
2. Let $G = (V, E)$ be a directed graph, in which every edge has capacity equal to 1 and some arbitrary cost. Edge costs could be positive, negative, or zero. Suppose you have just finished computing the minimum-cost circulation in this graph. Unfortunately, after all that work, now you realize that you recorded the direction of one of the edges incorrectly!

   Describe and analyze an algorithm to update the minimum-cost circulation in G when the direction of an arbitrary edge in G is reversed. The input to your algorithm consists of the directed graph G, the costs of edges in G, the minimum-cost circulation in G, and the edge to be reversed. Your algorithm should be faster than recomputing the minimum-cost circulation from scratch.

3. The ***chromatic number*** $\chi(G)$ of an undirected graph G is the minimum number of colors required to color the vertices, so that every edge has endpoints with different colors. Computing the chromatic number exactly is NP-hard, because 3COLOR is NP-hard.

   Prove that the following problem is also NP-hard: Given an arbitrary undirected graph G, return any integer between $\chi(G)$ and $\chi(G) + 473$.

4. Suppose you are given an $n \times n$ checkerboard with some of the squares deleted. You have a large set of dominos, just the right size to cover two squares of the checkerboard. Describe and analyze an algorithm to determine whether it is possible to tile the remaining squares with dominos—each domino must cover exactly two undeleted squares, and each undeleted square must be covered by exactly one domino.



Your input is a two-dimensional array $Deleted[1..n, 1..n]$ of bits, where $Deleted[i, j] = \text{TRUE}$ if and only if the square in row i and column j has been deleted. Your output is a single bit; you do ***not*** have to compute the actual placement of dominos. For example, for the board shown above, your algorithm should return TRUE.

5. Recall from Homework 11 that a ***prefix*** of a directed graph $G = (V, E)$ is a subset $P \subseteq V$ of the vertices such that no edge $u \to v \in E$ has $u \notin P$ and $v \in P$.

   Suppose you are given a rooted tree T, with all edges directed away from the root; every vertex in T has a weight, which could be positive, negative, or zero. Describe and analyze a *self-contained* algorithm to compute the prefix of T with maximum total weight. *[Hint: Don't use linear programming.]*

6. Suppose we are given a sequence of n linear inequalities of the form $a_i x + b_i y \le c_i$; the set of all points (x, y) that satisfy these inequalities is a convex polygon P in the (x, y)-plane. Describe a linear program whose solution describes the largest square with horizontal and vertical sides that lies inside P. (You can assume that P is non-empty.)