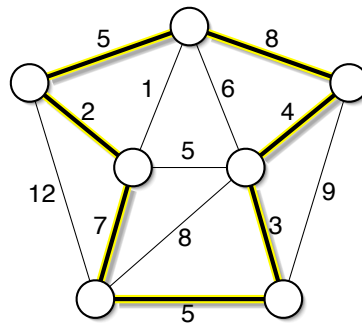


You have 180 minutes to answer six of these questions.
Write your answers in the separate answer booklet.
 You may take the question sheet with you when you leave.

1. Describe and analyze an algorithm that randomly shuffles an array $X[1..n]$, so that each of the $n!$ possible permutations is equally likely, in $O(n)$ time. (Assume that the subroutine $\text{RANDOM}(m)$ returns an integer chosen uniformly at random from the set $\{1, 2, \dots, m\}$ in $O(1)$ time.)
2. Let G be an undirected graph with weighted edges. A *heavy Hamiltonian cycle* is a cycle C that passes through each vertex of G exactly once, such that the total weight of the edges in C is at least half of the total weight of all edges in G . Prove that deciding whether a graph has a heavy Hamiltonian cycle is NP-complete.



A heavy Hamiltonian cycle. The cycle has total weight 34; the graph has total weight 67.

3. Suppose you are given a directed graph $G = (V, E)$ with capacities $c : E \rightarrow \mathbb{Z}_+$ and a maximum flow $F : E \rightarrow \mathbb{Z}$ from some vertex s to some other vertex t in G . Describe and analyze efficient algorithms for the following operations:
 - (a) $\text{INCREMENT}(e)$ — Increase the capacity of edge e by 1 and update the maximum flow F .
 - (b) $\text{DECREMENT}(e)$ — Decrease the capacity of edge e by 1 and update the maximum flow F .

Both of your algorithms should be significantly faster than recomputing the maximum flow from scratch.
4. Suppose you are given an undirected graph G and two vertices s and t in G . Two paths from s to t are *vertex-disjoint* if the only vertices they have in common are s and t . Describe and analyze an efficient algorithm to compute the maximum number of vertex-disjoint paths between s and t in G . [Hint: Reduce this to a more familiar problem on a suitable directed graph G' .]

5. A sequence of numbers $\langle a_1, a_2, a_3, \dots, a_n \rangle$ is *oscillating* if $a_i < a_{i+1}$ for every *odd* index i and $a_i > a_{i+1}$ for every *even* index i . For example, the sequence $\langle 2, 7, 1, 8, 2, 8, 1, 8, 3 \rangle$ is oscillating. Describe and analyze an efficient algorithm to compute the longest oscillating subsequence in a sequence of n integers.
6. Let $G = (V, E)$ be an undirected graph, each of whose vertices is colored either red, green, or blue. An edge in G is *boring* if its endpoints have the same color, and *interesting* if its endpoints have different colors. The *most interesting 3-coloring* is the 3-coloring with the maximum number of interesting edges, or equivalently, with the fewest boring edges. Computing the most interesting 3-coloring is NP-hard, because the standard 3-coloring problem we saw in class is a special case.
- (a) Let $zzz(G)$ denote the number of boring edges in the most interesting 3-coloring of a graph G . Prove that it is NP-hard to approximate $zzz(G)$ within a factor of $10^{10^{100}}$.
- (b) Let $wow(G)$ denote the number of interesting edges in the most interesting 3-coloring of G . Suppose we assign each vertex in G a *random* color from the set {red, green, blue}. Prove that the expected number of interesting edges is at least $\frac{2}{3}wow(G)$.
7. It's time for the 3rd Quasi-Annual Champaign-Urbana Ice Motorcycle Demolition Derby Race-O-Rama and Spaghetti Bake-Off! The main event is a competition between two teams of n motorcycles in a huge square ice-covered arena. All of the motorcycles have spiked tires so that they can ride on the ice. Each motorcycle drags a long metal chain behind it. Whenever a motorcycle runs over a chain, the chain gets caught in the tire spikes, and the motorcycle crashes. Two motorcycles can also crash by running directly into each other. All the motorcycle start simultaneously. Each motorcycle travels in a straight line at a constant speed until it either crashes or reaches the opposite wall—no turning, no braking, no speeding up, no slowing down. The Vicious Abscissas start at the south wall of the arena and ride directly north (vertically). Hell's Ordinates start at the west wall of the arena and ride directly east (horizontally). If any motorcycle completely crosses the arena, that rider's entire team wins the competition.
- Describe and analyze an efficient algorithm to decide which team will win, given the starting position and speed of each motorcycle.

