Homework 11

CS570 Spring 2025

Due: Apr 25, 2025

Ungraded Problems

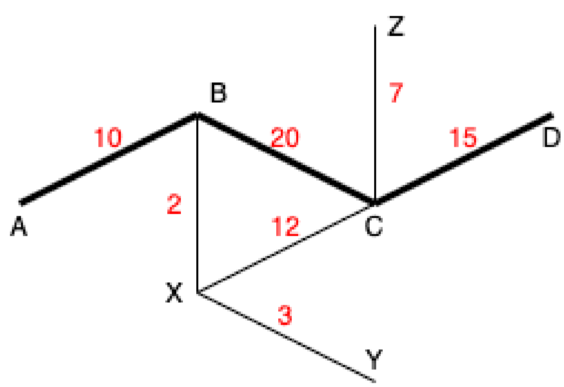
5. The government wants to build a multi-lane highway across the country. The plan is to choose a route and rebuild the roads along this route. We model this problem with a simple weighted undirected graph G with the nodes denoting the cities and the edges capturing the existing road network. The weight of an edge denotes the length of the road connecting the corresponding two cities (assumed positive).

   Let d_{uv} denote the shortest path distance between nodes u and v.

   Let $d(v, P)$ denote the shortest path distance from a node v to the *closest* node on a path P (i.e. $\min_{u \in P} d_{uv}$).

   Finally, we define the *aggregate remoteness* of P as $r(P) = \sum_{v \in V} d(v, P)$.

   In the example shown in the figure below, path $P = ABCD$ is a potential highway. Then, $d(A, P) = d(B, P) = d(C, P) = d(D, P) = 0$, while $d(X, P) = d_{XB} = 2$, $d(Y, P) = d_{YB} = 5$, and, $d(Z, P) = d_{ZC} = 7$. The remoteness of path ABCD is $0 + 0 + 0 + 0 + 2 + 5 + 7 = 14$.



   The government wants a highway with the minimum aggregate remoteness, so that all the cities are somewhat close to the highway. Formally, we state the problem as follows, "Given a graph G, and a non-negative number k, does there exist a path P in G having remoteness $r(P)$ at most k"? Show that this problem REMOTE-PATH is NP-complete.

   REMOTE-PATH $\in$ NP (6 points): A path will be a certificate (thus, poly-size). For each vertex v, $d(v, P)$ can be computed by first running Dijkstra from v and then finding the closest point on P. Adding $d(v, P)$ for all the nodes v gives the total remoteness. Thus, the certifier runs in polynomial time.

NP-hardness (14 points): We show this with a reduction from Hamiltonian Path (HP). Consider an HP instance graph $G = (V, E)$. Construct a weighted graph H by assigning arbitrary positive weights to all the edges in E. Pass this weighted graph H and $k = 0$ as input to REMOTE-PATH. We show that G has a hamiltonian path if and only if H has a path with zero remoteness. To prove the first direction, suppose P is a hamiltonian path of G. Since, all the nodes lie on P, we have $d(v, P) = 0$ $\forall v \in V$, and hence, $r(P) = 0$. Hence, REMOTE-PATH outputs 'Yes'. For the other direction, suppose H has a path with zero remoteness, say P. If there is a node $v \notin P$, then $d(v, P) > 0$ since all the edge-weights are positive. Hence, as $r(P) = 0$, P must contain all the nodes, thus, G indeed has a hamiltonian path. Since assigning the weights is done in $O(|E|)$ time, the reduction is poly-time.

6. For any positive k, the graph k-coloring problem is stated as follows: Determine if the vertices of a given graph G can be colored using k colors such that no two adjacent vertices have the same color. The graph 3-coloring problem is known to be NP-complete. Prove that the 5-coloring problem is NP-complete.

   Solution:

   First, we show that 5-coloring is in NP.

   Certificate: a color solution for the network, i.e., each node has a color.
   Certifier:

   - Check for each edge (u, v), the color of node u is different from the color of node v;
   - Check at most 5 colors are used

   This process takes polynomial time to verify that 5-coloring has a solution. 5-coloring is in NP.
   NP-hard: prove that 3-coloring $\leq_P$ 5-coloring.

   Graph construction:
   Given an arbitrary graph G. Construct G' by adding 2 new nodes u and v to G. Connect u and v to all nodes that existed in G, and to each other. G' can be colored with 5 colors iff G can be colored with 3 colors.

   (a) If there is valid 3-color solution for G, say using colors $1, 2, 3$, we want to show there is a valid 5-coloring solution to G'. We can color G' using five colors by assigning colors to G according to the 3-color solution, and then color node u and v by additional two different colors. In this case, node u and v have different colors from all the other nodes in G', and together with the 3-coloring solution in G, we use at most 5 colors to color G'.

   (b) If there is a valid 5-coloring solution for G', we want to show there is a valid 3-coloring solution in G. In G', since node u and v connect to all the other nodes in G and to each other, the 5-coloring solution must assign two different colors to node u and v, say colors 4 and 5. This leaves the remaining graph as G and only 3 remaining colors in our set. Since we do have a valid 5-coloring solution, the remaining 3 colors successfully form a valid 3-color solution for G.

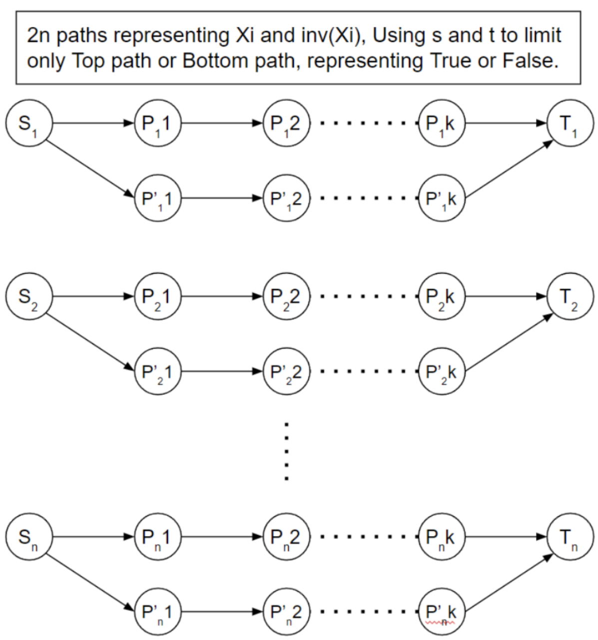
   Thus, 3-coloring $\leq_P$ 5-coloring.

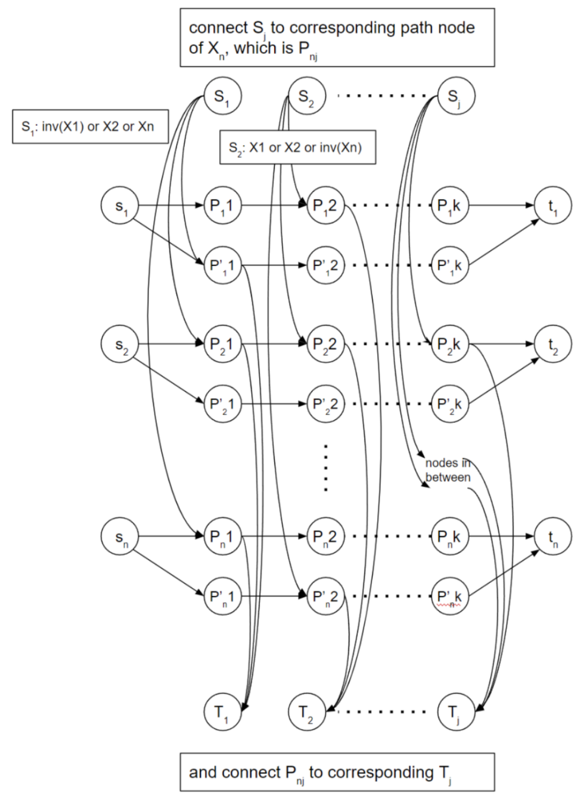
7. You are given a directed graph $G = (V, E)$ with weights on its edges $e \in E$. The weights can be negative or positive. The Zero-Weight-Cycle Problem is to decide if there is a simple cycle in G so that the sum of the edge weights on this cycle is exactly 0. Prove that this problem is NP-complete. Hint: For NP-hardness, reduce from the Subset sum problem that can be stated as "Given the set of positive numbers $S = \{w_1, ..., w_n\}$ and target sum $W > 0$, is there is a subset of S that adds up to exactly W ?".

   Solution: Zero-weight-cycle is in NP because we can exhibit a cycle in G, and it can be checked that the sum of the edge weights on this cycle are equal to 0.
   We now show that subset sum $\leq$ Zero-weight-cycle. We construct an instance of the Zero-weight-cycle in which the graph has nodes $0, 1, 2, ..., n$, and an edge (i, j) for all pairs $i < j$. The weight of the edge (i, j) is equal to w_j. Finally, there are edges $(j, 0)$ of weight $-W$ for all $j = 1, \ldots, n$.
   We claim that there is a subset that adds up to exactly W if and only if G has a zero-weight-cycle. If there is such a subset S, then we define a cycle that starts at 0, goes through the nodes whose indices are in S, and then returns to 0 on the edge $(n, 0)$. The weight of $-W$ on the edge $(n, 0)$ precisely cancels the sum of the other edge weights. Conversely, all cycles in G must use the edge $(n, 0)$, and so if there is a zero-weight-cycle, then the other edges must exactly cancel $-W$, in other words, their indices must form a set that adds up to exactly W.

8. The Directed Disjoint Paths Problem is defined as follows. We are given a directed graph G and k pairs of nodes $(s_1, t_1), (s_2, t_2), ..., (s_k, t_k)$. The problem is to decide whether there exist node-disjoint paths $P_1, P_2, ..., P_k$ so that P_i goes from s_i to t_i. Show that Directed Disjoint Paths is NP-complete.

Solution: The problem is in NP, since we can exhibit a set of disjoint paths P_i, and it can be checked in polynomial time that they are paths in G, connect the corresponding nodes, and are disjoint.
Now we show 3-SAT $\leq_p$ Directed Disjoint Paths. Consider a 3-SAT problem given by a set of clauses $C_1, ..., C_k$, each of length 3, over a set of variables $X = x_1, ..., x_n$. To create the corresponding instance of the Directed Disjoint Paths problem, we will have $2n$ directed paths, each of length k, one path P_i corresponding to variable x_i and one path P'_i corresponding to $\overline{x_i}$. We add n source-sink pairs corresponding to the n variables, and connect source s_i to the first node on paths P_i and P'_i and connect the last nodes in paths P_i and P'_i to sink t_i. Note that there are two directed paths connecting s_i to t_i : the path (s_i, P_i, t_i), and the path (s_i, P'_i, t_i). We will think of selecting the first of these paths as setting the variable x_i to false (as the variable through the copies of $\overline{x_i}$ are left unused), and selecting the second path will correspond to setting the variable x_i to true.





Now we will add k additional source sink pairs, one corresponding to each clause C_j.Let S_j and T_j be the source sink pair corresponding to clause C_j. We will claim that there is a path from S_j to T_j disjoint from the path selected to connect the $s_j - t_j$ source-sink pairs if and only if clause C_j is satisfied by the corresponding assignment. Assume clause C_j contains the literal t_{j1}, t_{j2} and t_{j3}. Now we have a path P_i or P'_i corresponding to each of these variables or negated variables. The paths have n nodes each, let v_{j1}, v_{j2} and v_{j3} denote the j-th node on the 3 corresponding paths. We add the edges (S_j, v_{jl}) and (v_{jl}, T_j) for each of $l = 1, 2, 3$.
Now we claim that the resulting directed graph has node disjoint paths connecting the source-sink pairs $s_i - t_i$ and $S_j - T_j$ for $i = 1, 2, ..., n$ and $j = 1, ..., k$ if and only if the 3-SAT instance is satisfiable. One direction is easy to see: if the 3-SAT instance is satisfable, then select the paths connecting s_i to t_i corresponding to the satisfying assignment, as suggested above. Then the source-sink pair S_j and

T_j can be connected through the path using the true variable in the clause.
Finally, we need to show that if the disjoint paths exist, then the 3-SAT formula has a satisfying assignment. Note that the paths P_i and P_j are disjoint, and the graph has no edges connecting different paths. The only edges outside these paths in the graph are edges entering one of the sinks, or leaving a source. As a result the only paths in the graph connecting a $s_i - t_i$ pair are the two paths (s_i, P_i, t_i) and (s_i, P'_i, t_i), and the only paths in G connecting $S_j - T_j$ pairs are the three possible paths through each of the 3 variable nodes in C. Hence, sets of disjoint paths connecting the source-sink pairs,correspond to satisfying assignments.