31 (100 PTS.) Walk with me

31.A. (50 PTS.) We are given a directed graph G with n vertices and m edges ($m \geq n$), where each vertex v has a height $h(v)$. The cost of traversing an edge (u, v) is defined as $c(u, v) = |h(v) - h(u)|$. The cost of a walk in G is the sum of the costs of edges in the walk. Prove that finding the least cost walk that visits all the vertices is NP-HARD. (In a walk, vertices and edges may be repeated. The start and end vertex may be different.)

Solution:

We will prove that the problem is NP-HARD via a reduction from the directed HAMILTONIAN-PATH. Given a directed graph G with n vertices and m edges, replace each vertex v with two vertices: v_{in} and v_{out}. Replace every edge (u, v) with (u_{out}, v_{in}). Finally, set the height: $h(v_{in}) = 0$ and $h(v_{out}) = 1$. We also add two special vertices. A vertex s of height 1, that has edges (s, x_{in}) connected to it, for all $x \in V(\mathsf{G})$, And a vertex t of height 0 that is connected by edges (x_{out}, t), for all $x \in V(G)$. The resulting graph G' has $2n + 2$ vertices and $3n + m$ edges and can be constructed in $O(n + m)$ time.

Claim 11.1. *The graph G has a directed Hamiltonian path $\iff$ the minimum cost walk that visits all the vertices in G' has cost $2n + 1$.*

Proof: $\implies$ Given a directed Hamiltonian path in G formed of vertices $v^1 \to v^2 \to \cdots \to v^n$, the corresponding walk in G' visits all the vertices of G' is $s \to v^1_{in} \to v^1_{out} \to v^2_{in} \to v^2_{out} \to \cdots \to v^n_{out} \to t$. The cost of this walk is $2n + 1$. The cost of all the edges in G' is 1 since an edge e is either of the form (u_{out}, v_{in}), (v_{in}, v_{out}), (s, v_{in}), or (v_{out}, t). All these edges cost 1. Hence, any walk that visits all the $2n + 2$ vertices in G' must cost at least $2n + 1$. Thus, we found a minimum cost walk that visits all the vertices in G'.

$\impliedby$ Assume you are given a walk σ that visits all the vertices in G' of cost $2n + 1$. The walk must start at s and ends at t, since these two vertices have only outgoing and incoming edges, respectively. Assume that

$$\sigma \equiv s \to v^1_{in} \to v^1_{out} \to v^2_{in} \to v^2_{out} \to \cdots \to v^n_{out} \to t.$$

And consider the corresponding walk in the original graph G'.

$$\begin{aligned}\pi &\equiv s \to v^1_{in} \to v^1_{out} \to v^2_{in} \to v^2_{out} \to \cdots \to v^n_{out} \to t\\ &= v^1 \to v^2 \to \cdots \to v^n.\end{aligned}$$

Clearly, π is a path of length $n - 1$ that visits all the vertices of G exactly once. That is, it is a Hamiltonian path of G, as claimed. ∎

As such, given an instance of Hamiltonian Path G, we convert it into G' as described above in polynomial time. We compute the cheapest walk that visits all the vertices of G' (i.e., we assume that we have a black box B that can solve this problem). If this walk has cost $2n + 1$, then by the above G has a Hamiltonian path, otherwise it doesn't.

The completes the reduction proof. To convince yourself that this is a correct reduction, assume that B works in polynomial time. If it did, then we just described a polynomial time algorithm that solves Hamiltonian Path in polynomial time. Hamiltonian Path is NP-Complete, which implies that given such a polynomial time B, all the problems in NP can be solved in polynomial time.

31.B. (50 pts.) We are given a directed graph G with n vertices and m edges ($m \geq n$), where each edge e has a set of colors $C(e) \subseteq \{1, ..., k\}$. Prove that deciding whether there exists a walk that uses all k colors (i.e. the union of the sets of colors of the edges of walk covers all colors.) is NP-Hard. (Hint: Reduce from Set Cover.)

Solution:

We show that the above problem is NP-Hard via a reduction from the set cover problem. Given a universe set U of size n, a family $F = \{F_1, \ldots, F_m\}$ of m subsets of U ($F_j \subseteq U$), and an integer k, the problem is to determine if there are k subsets whose union is U i.e. covers all the elements in the set U.

For $i = 1, \ldots, k$, create k directed diamond graphs. Each graph has a source vertex s_i and a sink vertex t_i. For each subset $F_j \in F$, create a vertex v_{ij} and add the directed edges (s_i, v_{ij}) and (v_{ij}, t_i). Then, connect the diamonds by adding the edges (t_i, s_{i+1}). Finally, for each edge $e = (s_i, v_{ij})$, set $C(e) = F_j$. Otherwise, set $C(e) = \emptyset$. The resulting graph G has $O(km)$ vertices and edges and $n = |U|$ colors. Hence, we can construct it in polynomial time.

Claim 11.2. *G has a walk that uses all n colors $\iff$ (U, F) has a set cover of k subsets.*

Proof: $\implies$ Given a walk in G that uses all $|U|$ colors, the walk can clearly be extended to start at s_1 and end at t_k while continuing to use all colors. Any walk in G that covers all colors starting at s_1 and ending at t_k must be of the form: $s_1 \to v_{1j} \to t_1 \to s_2 \to \ldots s_i \to v_{ij'} \to t_i \to \ldots s_k \to v_{kj''} \to t_k$. Since the edges (v_{ij}, t_i) and (t_i, s_{i+1}) have no colors, then there must be k sets that cover U corresponding to the colors of the k edges (s_i, v_{ij}).

$\impliedby$ Given a set cover of k subsets $S_1, \ldots, S_k$ where $S_i = F_j \in F$ we can find a walk in G that starts at s_1 and ends at t_k and covers all colors by choosing to go through the vertex v_{ij} corresponding to the ith subset $S_i = F_j$. Since, $\cup_{i=1}^{k} S_i = U$, the walk covers all colors.

Note: You have seen a similar problem before in 25.B where each set contained a single color and $k = 3$. Solving 25.B required keeping track of $2^k = 8$ states which was not NP-Hard since k was constant. However, for arbitrary k, the solution of 25.B is exponential in the input k.

32 (100 PTS.) Things are hard.

32.A. (20 PTS.) Suppose we have n prisoners $P_1, ..., P_n$ that we want to place in some disconnected blocks of a prison. Each prisoner is assigned to one blocks and will not be able to access other blocks. However, some prisoners are enemies and cannot be placed in the same block. Given integers n and k and a list of enemies for each of the n prisoners, we want to determine whether k blocks are sufficient to house all the prisoners? Prove that this problem is NP-HARD.

Solution:

This is nothing but a graph coloring problem. Given a graph G and an integer k, can the vertices of G be colored with k colors such that no two adjacent vertices share a color? We can reduce coloring to the prison assignment problem above. Each prisoner is a vertex in the graph. There is an edge between two vertices if the prisoners are enemies. Clearly, if there is a k-coloring of G, then we can assign the prisoner into k blocks where each color corresponds to a block. Similarly, if we can assign the prisoners into k blocks then there is a k-coloring where each block corresponds to a color. Hence, there is a k-coloring of the graph $\iff$ k blocks are sufficient to house all the prisoners.

Formally, given an instance of graph COLORING, which is a graph $\mathsf{G} = (\mathsf{V}, E)$ and a number k, we generate an instance of the Prisoners assignment problem, by treating each vertex of V as a prisoner, and an edge of G specifies that two prisoners are enemies, and let k be the number blocks allowed. Let H be the resulting instance of the Prisoners assignment problem. Clearly, H can be solved $\iff$ G can be colored by k colors. Thus implying that if we can solve the decision version of the Prisoners assignment problem in polynomial time, then we can solve COLORING in polynomial time. Since the Prisoners assignment problem is clearly in NP (i.e., given a valid assignment we can verify it in polynomial time), it follows that the decision version of this problem is NP-COMPLETE. The optimization version, where we ask for the smallest k that works, is thus NP-HARD.

32.B. (40 PTS.) Let G be an arbitrary directed weighted graph with n vertices and m edges such that no edge weight is zero (weights can be positive or negative). Prove that finding a *zero-length* (zero weight) Hamiltonian cycle in G is NP-COMPLETE.

Solution:

Let refer to the above problem as CHZero.

The problem CHZero is in NP. Given an instance $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ of the CHZero, a solution to it is a list L of n vertices corresponding to a directed Hamiltonian cycle in order. We check that $(L(i), L(i+1)) \in \mathsf{E}$ for $1 \leq i < n$ and $(L(n), L(1)) \in \mathsf{E}$. We also check that the sum of the weights of the $n+1$ edges is equal to zero. This takes polynomial time and will only accept iff there is a zero-length Hamiltonian cycle in G.

CHZero is NP-HARD. Given an instance of directed HAMILTONIAN-CYCLE, which is a directed graph G with n vertices, we reduce it to CHZero. To this end, replace a vertex x of G by two vertices x_{in} and x_{out}, where all the edges (u, x) get rewritten as (u, x_{in}), and all the edges (x, v) are replaced by (x_{out}, v). All the edges in this graph are assigned weight 1, except for the new edge (x_{in}, x_{out}), which is assigned weight $-n$. Let H be the resulting weighted graph.

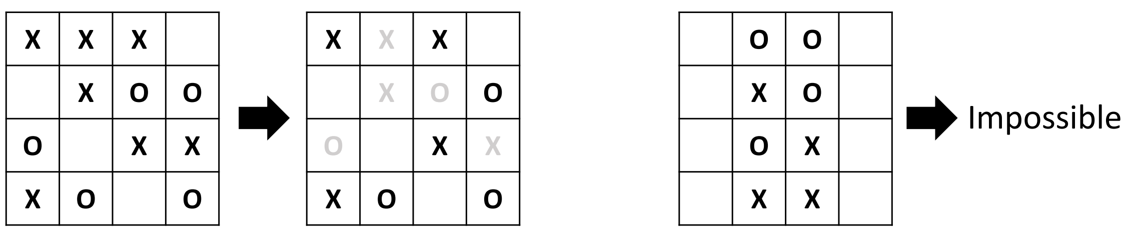
Claim 11.3. G *has a Hamiltonian cycle* $\iff$ H *has a zero-length Hamiltonian cycle.*

Proof: $\implies$ Given a Hamiltonian cycle C in G, then it can be modified to be a Hamiltonian cycle in H, by injecting the edge (x_{in}, x_{out}) into C where x appears. Clearly, this is a Hamiltonian cycle in H. It has n regular edges, of price 1, and one edge of price $-n$, which means that the total cost of the new Hamiltonian cycle C' of H is 0.

$\impliedby$ Given a zero length Hamiltonian cycle C' in H, it must include the edge (x_{in}, x_{out}) as this is the only edge with negative cost (and all other edges have a positive cost). As such, C' can be interpreted as a Hamiltonian cycle C in G by replacing (x_{in}, x_{out}) by x. ■

The above claim implies that the above reduction is correct, which implies that CHZero is NP-Complete.

32.C. (40 pts.) Consider the following **XO** puzzle. You are given an $n \times m$ grid of squares where each square has an **X**, an **O** or is empty. Your goal is to erase some of the **X**s and **O**s so that (1) every row contains at least one **X** or one **O** and (2) no column contains both **X** and **O**. For some input grids, it is impossible to solve the puzzle. The figure below shows two examples: a grid that is solvable and a grid that is impossible to solve. Prove that, given a grid, it is NP-Hard to determine whether the puzzle is solvable. (Hint: Reduce from 3SAT.)



Solution:

This problem is NP-Complete (not only NP-Hard). First, observe that is in NP– given a valid solution, one can easily verify that it complies with the given constraints in polynomial time.

Next, we prove the problem is NP-Hard by reducing 3SAT to determining whether the puzzle is solvable. Given a 3SAT formula φ with n variables $x_1, \ldots, x_n$ and m clauses $c_1, \ldots, c_m$, we construct an $m \times n$ **XO** puzzle. Each row corresponds to a clause and each column corresponds to a variable. If x_i is in clause c_j, place an **X** in cell at row j and column i. If $\overline{x_i}$ is in clause c_j, place an **O** in the cell at row j and column i. (We can assume that no clause has both x_i and $\overline{x_i}$ since $x_i \vee \overline{x_i} = 1$ and the clause can be ignored.) Let I be the resulting instance. The reduction from φ to I takes polynomial time since it is only filling up an $n \times m$ grid which is polynomial in the length of the 3SAT formula φ.

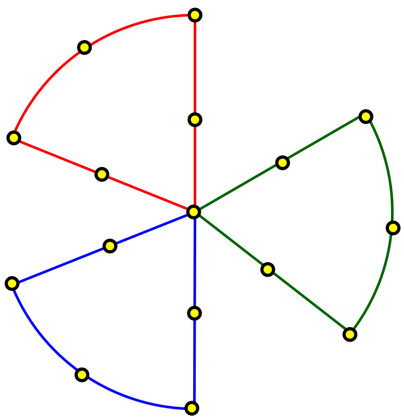
Claim 11.4. *The 3SAT formula* φ *is satisfiable* $\iff$ *the* **XO** *puzzle* I *is solvable.*

Proof: $\implies$ If φ is satisfiable, we can generate a satisfying assignment of the literals. If x_i is true, we erase all the **O**s from column i. If x_i is false, we erase all the **X**s from column i. Hence, only **X**s and **O**s corresponding to true literals remain and no column contains both **X** and **O**. Since the formula is satisfiable, every clause has at least one true literal i.e. every row has at least one **X** or one **O**. Thus, the puzzle is solvable.

$\Longleftarrow$ If the puzzle is solvable, we can take the solution. If row i contains **X**, set x_i to true. If row i contains **O**, set x_i to false. Since every row has at least one **X** or one **O**, then every clause has at least one true literal. Thus, the formula is satisfiable. ■

33 (100 PTS.) Fan

An undirected graph is a ***3-blade-fan*** if it consists of three cycles C_1, C_2, and C_3 of k nodes each and they all share exactly one node. Hence, the graph has $3k - 2$ nodes. The figure below shows a ***3-blade-fan*** of 16 nodes.



Given an undirected graph G with n vertices and m edges and an integer k, the FAN problem asks whether or not there exists a subgraph of G which is a ***3-blade-fan***. Prove that FAN is NP-COMPLETE.

Solution:

The problem is in NP. We let the certificate be three lists of length $k+1$ of vertices where each list corresponds to the vertices of one of the cycles in order and the lists begin and end with the vertex that is in common between the three cycles. We can verify that the lists form a ***3-blade-fan*** by performing the following checks:

- All 3 lists have $k+1$ vertices that are in G.
- All 3 lists start and end with the same vertex v.
- The intersection of any two lists is the same single vertex v.
- Any two consecutive vertices in any list have an edge between them in G.

Clearly, if the certificate passes all the checks, then one can form a $3k-2$ vertex subgraph that is a ***3-blade-fan*** from the three lists. Similarly, if there is a $3k-2$ vertex subgraph that is a ***3-blade-fan***, then one can provide three lists that will pass all the checks. Finally, the certificate is of polynomial length and the checks can be done in at most polynomial time.

The problem is NP-HARD by reduction from HAMILTONIAN-CYCLE to FAN. Given a graph G with n vertices, pick any vertex v in G and construct a new graph G' by adding two sets of $n-1$ degree 2 vertices to form two new cycles containing v. The reduction is polynomial time since we only added a linear number of vertices and edges to the graph.

Claim 11.5. G *has a Hamiltonian cycle* $\iff$ G' *contains a subgraph of* $3n - 2$ *nodes which is a* **3-blade-fan**.

Proof: $\implies$ If G has a Hamiltonian cycle, then we can form a ***3-blade-fan*** in G' by taking the Hamiltonian cycle along with the two new cycles of $n - 1$ vertices which we added. The three cycles is each of length n and share a single vertex v.

$\impliedby$ If G' has a $3n - 2$ vertex subgraph that is a ***3-blade-fan***, then this sub-graph uses all the vertices of G'. Each of the newly added vertices can be in only one of the two newly added cycles of length n that contain v. Thus, the third cycle (blade) of the fan must use all the vertices of G. Hence, it is a Hamiltonian cycle for G. ■

BTW, a cute alternative reduction (suggested by students at office hours) is to take the graph G and glue three copies of it at a common vertex. It is easy to verify that the resulting graph has a 3-blade $\iff$ the original graph has a Hamiltonian cycle.