CS/ECE 374 A ✧ Spring 2026

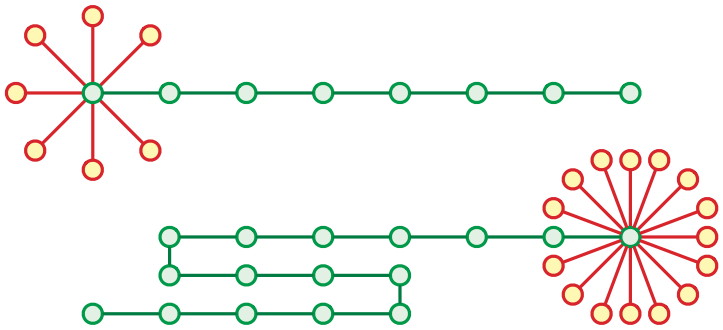
Homework 11

Due Tuesday, December 1, 2026 at 9pm Central Time

This is the last graded homework before the final exam.

---

1. (a) A ***dandelion of length ℓ*** consists of a path of length ℓ, with exactly ℓ new edges attached to one end. Prove that it is NP-hard to decide, given an undirected graph $G = (V, E)$ and an integer ℓ, whether G has a dandelion of length at least ℓ as a subgraph.



Two dandelions, one of length 7 and the other of length 15.

(b) A subset S of vertices in an undirected graph G is ***triangle-free*** if, for every triple of distinct vertices $u, v, w \in S$, at least one of the three edges uv, uw, and vw is *absent* from G. Prove that it is NP-hard to decide, given an undirected graph $G = (V, E)$ and an integer k, whether G has a triangle-free subset of k vertices.

2. You thought you could get away, but no. You've been brought back to your old game development studio for one last job: developing another sequel to the *Murder the Mountain* game, this time titled *Murder the Mountain* (December 2026). Yet again, each map can be modeled as a directed acyclic graph (dag) $G = (V, E)$ with a start vertex $s \in V$ and a boss vertex $b \in V$. The player character (PC) needs to follow a path from s to b. If the PC enters a vertex v, they will take some $Damage(v) \geq 0$ damage. The PC starts with a variable amount of health $k \geq 1$ and will die if they reach 0 health or less.

   Not content with using very large integers as the damage numbers, the creative leads behind the game decide each relic should *also* have a variable worth. Instead of the old Boolean-valued *Relic* function, maps now come with an *integer*-valued function $Worth : V \to \mathbb{N}$ giving the total worth of relics on each vertex. The ***total score*** of a path $P = \langle s = v_1, v_2, \ldots, v_\ell = b \rangle$ from s to b is $\sum_{v \in P} Worth(v)$ if the PC survives path P and $-\infty$ otherwise.

   Of course, each value *Worth* is *also* a very large integer. You sigh loudly and get to work.

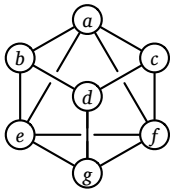
   (a) Prove that it is NP-hard to decide, given a dag $G = (V, E)$, two vertices $s, b \in V$, functions $Damage : V \to \mathbb{N}$ and $Worth : V \to \mathbb{N}$ as described above, starting health k, **and a target score** T, whether there is a path from s to b of total score **at least** T.

   (b) Another developer discovers some old testing code from the classic *Murder the Mountain* (November 2026). They get really excited, because the code can be modified easily to compute the maximum total score for any s to b path in only $O(k(|V| + |E|))$ time.[1] They claim the two of you have found a proof that P = NP, but that is, unfortunately, *not* true. Why not?

[1] Some code from the critically acclaimed *Final Murder the Mountain Quest XII* can be modified to compute the maximum total score in $O(r(|V| + |E|))$ time where $r = \sum_{v \in V} Worth(v)$.

Solved Problem

3. A *double-Hamiltonian tour* in an undirected graph G is a closed walk that visits every vertex in G exactly twice. Prove that it is NP-hard to decide whether a given graph G has a double-Hamiltonian tour.



This graph contains the double-Hamiltonian tour $a \to b \to d \to g \to e \to b \to d \to c \to f \to a \to c \to f \to g \to e \to a$.

Solution: We prove the problem is NP-hard with a reduction from the standard Hamiltonian cycle problem. Let G be an arbitrary undirected graph. We construct a new graph H by attaching a small gadget to every vertex of G. Specifically, for each vertex v, we add two vertices $v^\sharp$ and $v^\flat$, along with three edges $vv^\flat$, $vv^\sharp$, and $v^\flat v^\sharp$.



A vertex in G and the corresponding vertex gadget in H.

Now I claim that

G has a Hamiltonian cycle if and only if H has a double-Hamiltonian tour.

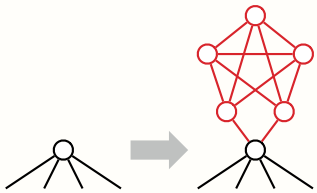
$\Longrightarrow$ Suppose G contains a Hamiltonian cycle $C = (((v_1, v_2), \cdots), (v_n, v_1))$. We can construct a double-Hamiltonian tour of H by replacing each vertex v_i in C with the following walk:

$$(((\cdots, v_i), (v_i^\flat, v_i^\sharp)), ((v_i^\flat, v_i^\sharp), (v_i, \cdots)))$$

$\Longleftarrow$ Conversely, suppose H has a double-Hamiltonian tour D. Consider any vertex v in the original graph G; the tour D must visit v exactly twice. Those two visits split D into two closed walks, each of which visits v exactly once. Any walk from $v^\flat$ or $v^\sharp$ to any other vertex in H must pass through v. Thus, one of the two closed walks visits only the vertices v, $v^\flat$, and $v^\sharp$. Thus, if we remove the vertices and edges in $H \setminus G$ from D, we obtain a closed walk in G that visits every vertex in G exactly once.

Given any graph G, we can clearly construct the corresponding graph H in polynomial time by brute force.

With more effort, we can construct a graph H that contains a double-Hamiltonian tour ***that traverses each edge of H at most once*** if and only if G contains a Hamiltonian cycle. For each vertex v in G we attach a more complex gadget containing five vertices and eleven edges, as shown on the next page.



A vertex in G, and the corresponding modified vertex gadget in H.

■

Rubric: 10 points, standard polynomial-time reduction rubric. This is not the only correct solution.

Non-solution (self-loops): We attempt to prove the problem is NP-hard with a reduction from the Hamiltonian cycle problem. Let G be an arbitrary undirected graph. We construct a new graph H by attaching a self-loop every vertex of G. Given any graph G, we can clearly construct the corresponding graph H in polynomial time.



An incorrect vertex gadget.

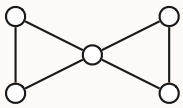
Now I claim that

G has a Hamiltonian cycle
if and only if
H has a double-Hamiltonian tour.

$\Longrightarrow$ Suppose G has a Hamiltonian cycle $(((v_1, v_2), \cdots), (v_n, v_1))$. We can construct a double-Hamiltonian tour of H by alternating between edges of the Hamiltonian cycle and self-loops: $v_1 \to v_1 \to v_2 \to v_2 \to v_3 \to \cdots \to v_n \to v_n \to v_1$.

$\not\Longleftarrow$ Um. . .

Unfortunately, if H has a double-Hamiltonian tour, we *cannot* conclude that G has a Hamiltonian cycle, because we cannot guarantee that a double-Hamiltonian tour in H uses *any* self-loops. The graph G shown below is a counterexample; it has a double-Hamiltonian tour (even before adding self-loops!) but no Hamiltonian cycle.



This graph has a double-Hamiltonian tour.

♣

Some useful NP-hard problems. You are welcome to use any of these in your own NP-hardness proofs, except of course for the specific problem you are trying to prove NP-hard.

3SAT: Given a boolean formula in conjunctive normal form, with exactly three distinct literals per clause, does the formula have a satisfying assignment?

INDEPENDENTSET: Given an undirected graph G and an integer k, is there a subset of k vertices in G that have no edges among them?

CLIQUE: Given an undirected graph G and an integer k, is there a subset of k vertices in G where there is an edge between every pair of them?

VERTEXCOVER: Given an undirected graph G and an integer k, is there a subset of k vertices that touch every edge in G?

DOMINATINGSET: Given an undirected graph G and an integer k, is there a subset of k vertices such that every vertex of G is either in the set or adjacent to a member of the set?

SETCOVER: Given a collection of subsets $S_1, S_2, \ldots, S_m$ of a set S and an integer k, is there a subcollection of k of these subsets whose union is S?

HITTINGSET: Given a collection of subsets $S_1, S_2, \ldots, S_m$ of a set S and an integer k, is there a subset of S of size k that intersects every subset S_i?

3COLOR: Given an undirected graph G, can its vertices be colored with three colors so that every edge touches vertices with two different colors?

HAMILTONIANPATH: Given graph G (either directed or undirected), is there a path in G that visits every vertex exactly once?

HAMILTONIANCYCLE: Given a graph G (either directed or undirected), is there a cycle in G that visits every vertex exactly once?

TRAVELINGSALESMAN: Given a graph G (either directed or undirected) with weighted edges and a number L, is there a Hamiltonian path/cycle of weight at most L in G?

LONGESTPATH: Given a graph G (either directed or undirected, possibly with weighted edges) and a number L, is there a simple path of length at least L in G?

STEINERTREE: Given an undirected graph G with some of the vertices marked and an integer k, is there a subtree of G with at most k edges that contains every marked vertex?

SUBSETSUM: Given a set X of positive integers and an integer k, does X have a subset whose elements sum to k?

PARTITION: Given a set X of positive integers, can X be partitioned into two subsets with the same sum?

3PARTITION: Given a set X of $3n$ positive integers, can X be partitioned into n three-element subsets, all with the same sum?

DRAUGHTS: Given an $n \times n$ international draughts configuration and an integer k, is there a move that can (and therefore must) capture at least k pieces in a single move?