Write your answers in the separate answer booklet.

You have 180 minutes (after you get the answer booklet) to answer seven questions.

1. For each statement below, there are two boxes in the answer booklet labeled "Yes" and "No". Check "Yes" if the statement is ***always*** true and "No" otherwise, and give a ***brief*** (at most one short sentence) explanation of your answer. **Assume $P \neq NP$.** If there is any other ambiguity or uncertainty about an answer, check "No". For example:

   - $x + y = 5$

     [Yes] [X] No — Suppose $x = 3$ and $y = 4$.

   - 3SAT can be solved in polynomial time.

     [Yes] [X] No — 3SAT is NP-hard.

   - If P = NP, then Emily can beat Super Mario Brothers in four minutes and 53 seconds.

     [X] Yes [ ] No — The hypothesis is false, so the implication is true.

   Read each statement *very* carefully; some of these are deliberately subtle!

   (a) Which of the following statements are true for ***every*** language $L \subseteq \{0,1\}^*$?

   - $(L^*)^*$ is infinite.
   - If L is decidable then its complement $\overline{L}$ is undecidable.
   - $\{\langle M \rangle \mid M \text{ accepts } L\}$ is undecidable.
   - Either L is finite or L is NP-hard. (The *or* is inclusive.)
   - Either L has an infinite fooling set or $L \in P$. (The *or* is inclusive.)

   (b) Consider the following pair of languages:

   - TREE $= \{G \mid G \text{ is a connected undirected graph with no cycles}\}$
   - HAMPATH $= \{G \mid G \text{ is an undirected graph that contains a Hamiltonian path}\}$

   (For concreteness, assume that in both of these languages, graphs are represented by their adjacency matrices.) Which of the following statements are true, assuming P $\neq$ NP?

   - TREE is NP-hard.
   - TREE $\cap$ HAMPATH is NP-hard.
   - TREE $\cup$ HAMPATH is NP-hard.
   - HAMPATH is undecidable.
   - A reduction from TREE to HAMPATH would imply P = NP.

   ↖ no time stated

Problems 2–7 appear on the next three pages.

2. Suppose we want to prove that the following language is undecidable.

$$\textsc{Chalmers} := \{\langle M \rangle \mid M \text{ accepts both } \texttt{STEAMED} \text{ and } \texttt{HAMS}\}$$

(a) Professor Skinner suggests a reduction from the standard halting language

$$\textsc{Halt} := \{(\langle M \rangle, w) \mid M \text{ halts on inputs } w\}.$$

Specifically, Professor Skinner claims that if there is a Turing machine SuperNintendo that decides the language Chalmers, then the following algorithm decides Halt.

```
DecideHalt((⟨M⟩, w)):
  Encode the following Turing machine:
    AuroraBorealis(x):
      if x = STEAMED or x = HAMS or x = UTICA
        run M on input w
        return False ⟨⟨reject x⟩⟩
      else
        return True ⟨⟨accept x⟩⟩
  return SuperNintendo(⟨AuroraBorealis⟩)
```

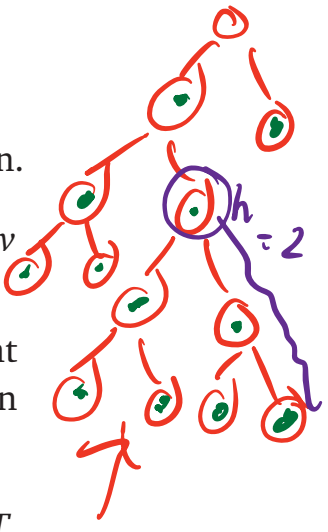
Show that Professor Skinner's reduction is incorrect by describing a Turing machine (or an algorithm) M and string w such that $\textsc{DecideHalt}(\langle M \rangle, w))$ either never returns or returns the wrong answer.

(b) Correctly ***prove*** that Chalmers is undecidable.

3. Let T be a *full* binary tree, meaning that every node has either two children or no children.

- Recall that the *height* of a vertex v in T is the length of the longest path in T from v down to a leaf. In particular, every leaf of T has height zero.
- A vertex v is *AVL-balanced* if v is a leaf, or if the heights of v's children differ by at most 1. (You might recall from CS 225 that an ***AVL-tree*** is a binary search tree in which *every* vertex is AVL-balanced.)

Describe and analyze an algorithm to compute the number of AVL-balanced vertices in T.



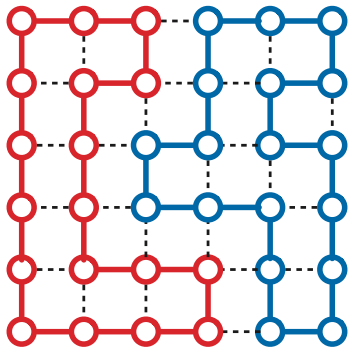
recursion?

DP?

Problems 4–7 appear on the next two pages.

4. Submit a solution to ***exactly one*** of the following problems.

   (a) A *Hamiltonian bicycle* in a graph G is a pair of simple cycles in G, with identical lengths, such that every vertex of G lies on exactly one of the two cycles.



A Hamiltonian bicycle in the 6×6 grid graph.

   Prove that determining whether a given undirected graph G has a Hamiltonian bicycle is NP-*complete*. from Ham cycle?

   (b) A *clique-partition* of a graph $G = (V, E)$ is a partition of the vertices V into disjoint subsets $V_1 \cup V_2 \cup \cdots \cup V_k$, such that for each index i, every pair of vertices in subset V_i is connected by an edge in G. The *size* of a clique partition is the number of subsets V_i.

   Prove that determining whether a given undirected graph G has a clique-partition of size 3 is NP-*complete*. 3Color?

In fact, both of these problems are NP-complete, but we only want a proof for one of them. Don't forget to tell us which problem you've chosen!

5. Suppose we are given a directed graph $G = (V, E)$, where every edge $e \in E$ has a *positive* weight $w(e)$, along with two vertices s and t.

   (a) Suppose each *vertex* of G is colored either orange, green, or purple. Describe and analyze an algorithm to find the shortest walk from s to t in G that never visits two consecutive *vertices* with the same color.

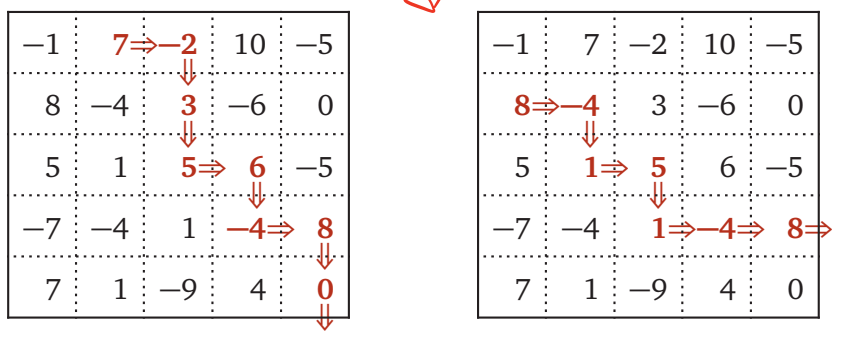
   (b) Now suppose each *edge* of G is colored either orange, green, or purple. Describe and analyze an algorithm to find the shortest walk from s to t in G that never traverses two consecutive *edges* with the same color.

graph reductions

Problems 5 and 6 appear on the next page.

6. *Vankin's Mile* is a solitaire game played on an $n \times n$ square grid. The player starts by placing a token on any square of the grid. Then on each turn, the player moves the token either one square to the right or one square down. The game ends when player moves the token off the edge of the board. Each square of the grid has a numerical value, which could be positive, negative, or zero. The player starts with a score of zero; whenever the token lands on a square, the player adds its value to his score. The object of the game is to score as many points as possible.

   For example, given the grid below, we can score $7-2+3+5+6-4+8+0=23$ points by following the path on the left, or we can score $8-4+1+5+1-4+8=15$ points by following the path on the right.



   (a) Describe and analyze an efficient algorithm to compute the maximum possible score for a game of Vankin's Mile, given the $n \times n$ array of values as input.

   (b) A variant called *Vankin's Niknav* adds an additional constraint to Vankin's Mile: *The sequence of values that the token touches must be a* ***palindrome***. Thus, the example path on the right is valid, but the example path on the left is not. Describe and analyze an efficient algorithm to compute the maximum possible score for an instance of Vankin's Niknav, given the $n \times n$ array of values as input.

7. (a) Let L_a denote the set of all strings $w \in \{0,1,2\}^*$ such that $\#(1,w) + 2 \cdot \#(2,w)$ is divisible by 3. For example, L_a contains the strings `0012` and `20210202` and the empty string ε, but L_a does not include the strings `121` or `0122210`.

   Describe a DFA or NFA that accepts L_a. (You do not need to prove that your answer is correct.)

   (b) Let L_b denote the set of all strings $w \in \{0,1,2\}^*$ such that no two symbols appear the same number of times, or in other words, the integers $\#(0,w)$ and $\#(1,w)$ and $\#(2,w)$ are all different. For example, L_b contains the strings `110212` and `20220`, but L_b does not include the strings `01212` or `2120210` or the empty string ε.

   Prove that L_b is not a regular language.

CS/ECE 374 A ✧ Spring 2026

Practice Final Exam 1

May 5, 2026

Name:	Emily Fox
NetID:	elcfox

- ***Don't panic!***
- You have 180 minutes to answer seven questions. The questions are described in more detail in a separate handout.
- If you brought anything except your writing implements, your two **hand-written** double-sided 8½" × 11" cheat sheets, and your university ID, please put it away for the duration of the exam. In particular, please turn off and put away *all* medically unnecessary electronic devices.
- Please clearly print your name and your NetID in the boxes above.
- Please also print your name at the top of every page of the answer booklet, except this cover page. We want to make sure that if a staple falls out, we can reassemble your answer booklet. (It doesn't happen often, but it does happen.)
- Greedy algorithms require formal proofs of correctness to receive any credit, even if they are correct. Otherwise, proofs or other justifications beyond items listed in the standard rubrics are required for full credit if and only if we explicitly ask for them, using the word ***prove*** or ***justify*** in bold italics.

- **Please do not write outside the black boxes on each page.** These indicate the area of the page that our scanners can actually scan. If the scanner can't see your work, we can't grade it.
- If you run out of space for an answer, please use the overflow/scratch pages at the back of the answer booklet, but **please clearly indicate where we should look**. If we can't find your work, we can't grade it.
- **Only work that is written into the stapled answer booklet will be graded**. In particular, you are welcome to detach scratch pages from the answer booklet, but any work on those detached pages will not be graded. We will provide additional scratch paper on request, but any work on that scratch paper will not be graded.
- Please return ***all*** paper with your answer booklet: your question sheet, your cheat sheet, and all scratch paper. **Please put all loose paper *inside* your answer booklet.**
- Breathe in. Breathe out. You've got this.

Announcements:

- FLEX closes Thu May 7 (Reading Day)
- Final Practice 3 today
- one "review party" Sunday 2-4pm (location TBA)
- office hours
- CA applications are open for Fall 2026 (by July 15)
- Final: May 14 (Thu) Conflict? (May 13?)

This sentence contains four and three fourths percent a's, five and one half percent c's, three percent d's, eighteen and one fourth percent e's, four and one half percent f's, three fourths percent g's, five percent h's, two and one half percent i's, two percent l's, twelve and one half percent n's, six percent o's, five percent p's, eight percent r's, seven percent s's, nine and three fourths percent t's, two and one fourth percent u's, one and one half percent v's, one and one fourth percent w's and one half percent x's.

CS/ECE 374 A ✧ Spring 2026 Practice Final 1 Problem 1	Name: Emily Fox

For each statement below, there are two boxes in the answer booklet labeled "Yes" and "No". Check "Yes" if the statement is ***always*** true and "No" otherwise, and give a ***brief*** (at most one short sentence) explanation of your answer. **Assume *P* ≠ *NP*.** If there is any other ambiguity or uncertainty about an answer, check "No".

Read each statement *very* carefully; some of these are deliberately subtle!

(a) Which of the following statements are true for ***every*** language $L \subseteq \{0,1\}^*$?

- $(L^*)^*$ is infinite.

  Yes ☐ No ☒ $L = \emptyset$ $(L^* = \{\epsilon\}, (L^*)^* = \{\epsilon\})$

- If L is decidable then its complement $\overline{L}$ is undecidable.

  Yes ☐ No ☒ $L = \emptyset$

- $\{\langle M \rangle \mid M \text{ accepts } L\}$ is undecidable.

  Yes ☐ No ☒ $L = Diverge$ (No Y machine for Rice's)

- Either L is finite or L is NP-hard. (The *or* is inclusive.)

  Yes ☐ No ☒ $L = \Sigma^*$

- Either L has an infinite fooling set or $L \in P$. (The *or* is inclusive.)

  Yes ☒ No ☐ if no inf. fooling set, L is regular. ($\forall$ has a DFA)

Problem 1 continues onto the next page.

(b) Consider the following pair of languages:

- TREE = {G | G is a connected undirected graph with no cycles}
- HAMPATH = {G | G is an undirected graph that contains a Hamiltonian path}

(For concreteness, assume that in both of these languages, graphs are represented by their adjacency matrices.) Which of the following statements are true, assuming P ≠ NP?

- TREE is NP-hard.

  - or · Tree ∈ P
  - why?

  Yes [ ] No [X] DFS + check if all marked + exactly n-1 edges

- TREE ∩ HAMPATH is NP-hard.

  it is a

  Yes [ ] No [X] in P: check if ∨ path

- TREE ∪ HAMPATH is NP-hard.

  Yes [X] No [ ] use normal HamPath reduction

- HAMPATH is undecidable.

  Yes [ ] No [X] Check all permutations of vertices

- A reduction from TREE to HAMPATH would imply P = NP.

  Yes [ ] No [X] P≠NP / no "poly time" / wrong direction

  pick one

CS/ECE 374 A ✧ Spring 2026 Practice Final 1 Problem 2	Name: Emily Fox

Suppose we want to prove that the following language is undecidable.

$$\textsc{Chalmers} := \{\langle M \rangle \mid M \text{ accepts both } \texttt{STEAMED} \text{ and } \texttt{HAMS}\}$$

(a) Show that Professor Skinner's reduction from HALT (described in the question sheets) is incorrect by describing a Turing machine (or an algorithm) M and string w such that $\textsc{DecideHalt}(\langle M \rangle, w))$ either never returns or returns the wrong answer.

(b) Correctly ***prove*** that CHALMERS is undecidable.

(a) M: machine that ignores input & immediately halts

w: ε

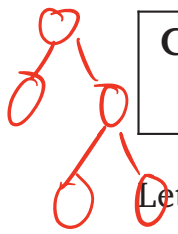
(AB will reject x = STEAMED, so <AB> ∉ Chalmers even though (<M>, w) ∈ Halt.)

(b) Rice's (acceptance) theorem

L: any language containing STEAMED & HAMS

Y: accept everything

N: reject everything



CS/ECE 374 A ✧ Spring 2026 Practice Final 1 Problem 3	Name: Emily Fox

Let T be a *full* binary tree, meaning that every node has either two children or no children.

- Recall that the *height* of a vertex v in T is the length of the longest path in T from v down to a leaf. In particular, every leaf of T has height zero.

recursion

- A vertex v is *AVL-balanced* if v is a leaf, or if the heights of v's children differ by at most 1. (You might recall from CS 225 that an ***AVL-tree*** is a binary search tree in which *every* vertex is AVL-balanced.)

Describe and analyze an algorithm to compute the number of AVL-balanced vertices in T.

---

height(v): height of vertex v.

$$\text{height}(v) = \begin{cases} 0 & \text{if } v \text{ is a leaf} \\ 1 + \max\{\text{height}(w) \mid w \text{ is a child of } v\} \end{cases}$$

Store heights as an extra entry on each vertex computed in postorder. (Stored as v.height for each vertex v.)

Algorithm:

Compute v.height for each vertex as described above.

```
total ← 0
for each vertex v
    if v is a leaf
        total ← total + 1
    else
        w1, w2 ← children of v
        if |w1.height - w2.height| ≤ 1
            total ← total + 1
return total
```

Time: O(n) where n is # vertices

CS/ECE 374 A ✧ Spring 2026 Practice Final 1 Problem 4	Name: Emily Fox

Submit a solution to ***exactly one*** of the following problems.

(a) A *Hamiltonian bicycle* in a graph G is a pair of simple cycles in G, with identical lengths, such that every vertex of G lies on exactly one of the two cycles.

Prove that determining whether a given undirected graph G has a Hamiltonian bicycle is NP-*complete*.

(b) A *clique-partition* of a graph $G = (V, E)$ is a partition of the vertices V into disjoint subsets $V_1 \cup V_2 \cup \cdots \cup V_k$, such that for each index i, every pair of vertices in subset V_i is connected by an edge in G. The *size* of a clique partition is the number of subsets V_i.

Prove that determining whether a given undirected graph G has a clique-partition of size 3 is NP-*complete*.

In fact, both of these problems are NP-complete, but we only want a proof for one of them. Don't forget to tell us which problem you've chosen!

(a) In NP: For Yes inputs use the two cycles (given as ~~disjoint~~ permutations of ~~the~~ two subset of vertices with each vertex included once) as certificate.

NP-hard. Reduce from Hamiltonian Cycle.

Given an undirected graph $G = (V, E)$, build a new undirected graph $G' = (V', E')$.

G' is two disjoint copies of G.

i.e. $V' := V \times \{0, 1\}$

$E' := \{(u,0)-(v,0) \mid u-v \in E\} \cup \{(u,1)-(v,1) \mid u-v \in E\}$

Takes polynomial $O(|V| + |E|)$ time.

cont. on page 8.

CS/ECE 374 A ✧ Spring 2026 Practice Final 1 Problem 5	Name: Emily Fox

Suppose we are given a directed graph $G = (V, E)$, where every edge $e \in E$ has a *positive* weight $w(e)$, along with two vertices s and t.

(a) Suppose each *vertex* of G is colored either orange, green, or purple. Describe and analyze an algorithm to find the shortest walk from s to t in G that never visits two consecutive *vertices* with the same color.

(b) Now suppose each *edge* of G is colored either orange, green, or purple. Describe and analyze an algorithm to find the shortest walk from s to t in G that never traverses two consecutive *edges* with the same color.

(a) Build graph $G' = (V, E')$.

$E' := \{(u,v) \mid (u,v) \in E$ & u & v are not the same colors$\}$

Want shortest walk from s to t in G'.

Use Dikstra's algorithm from s & return its path from s to t.

Time: $O(|E| \log |V|)$

(b) Build graph $G' = (V', E')$. not necessary

$V' := V \times \{0, G, P\}$ ← (last edge used)

$E' := \{((u,c),(v,c')) \mid (u,v) \in E,$ color of $(u,v) \neq c,$ c' = color of $(u,v)\}$

cont. of page 9

CS/ECE 374 A ✧ Spring 2026 Practice Final 1 Problem 6	Name: Emily Fox

(a) Describe and analyze an efficient algorithm to compute the maximum possible score for a game of Vankin's Mile, given the $n \times n$ array of values as input. *(See the question handout for a detailed description of Vankin's Mile.)*

(b) A variant called *Vankin's Niknav* adds an additional constraint: *The sequence of values that the token touches must be a* ***palindrome****.* Describe and analyze an efficient algorithm to compute the maximum possible score for an instance of Vankin's Niknav, given the $n \times n$ array of values as input.

Expect input as array $V[1..n, 1..n]$ where $V[i,j]$ is number i from top, j from left.

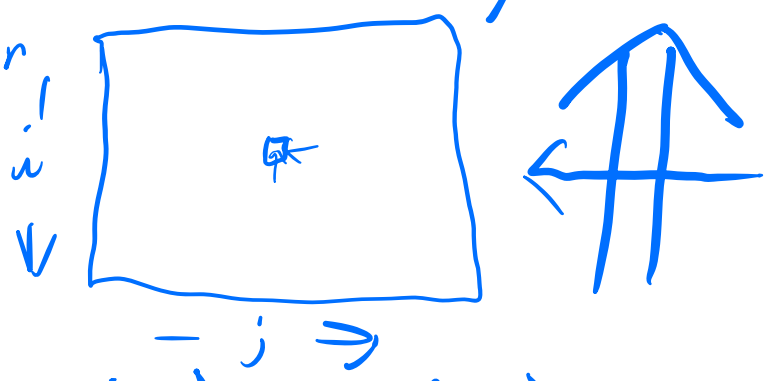
(a) $Mile(i,j)$: Max score possible starting with token at position (i,j). (i or $j = n+1$ means no game or a score of 0)

Want to compute $\max\{Mile(i,j) \mid 1 \le i \le n,\ 1 \le j \le n\}$.

$$Mile(i,j) = \begin{cases} 0 & i > n \text{ or } j > n \\ V[i,j] + \max\{Mile(i+1,j), Mile(i,j+1)\} & \text{o.w.} \end{cases}$$

Memoize in a 2D array $Mile[1..n+1, 1..n+1]$

Eval order



-or-

for i going down
for j going down

Time $O(1) \cdot O(n^2) = \underline{O(n^2)}$

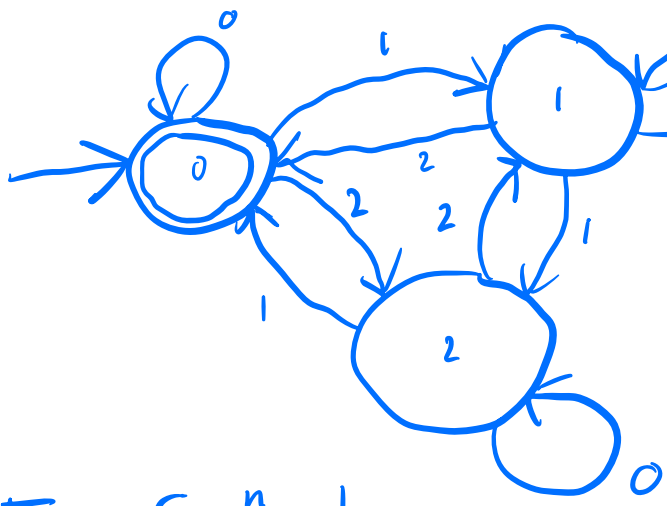
(6) on page 10

CS/ECE 374 A ✧ Spring 2026 Practice Final 1 Problem 7	Name: Emily Fox

(a) Let L_a denote the set of all strings $w \in \{0,1,2\}^*$ such that $\#(1,w) + 2 \cdot \#(2,w)$ is divisible by 3. Describe a DFA or NFA that accepts L_a. (You do not need to prove that your answer is correct.)

(b) Let L_b denote the set of all strings $w \in \{0,1,2\}^*$ such that no two symbols appear the same number of times, or in other words, the integers $\#(0,w)$ and $\#(1,w)$ and $\#(2,w)$ are all different. ***Prove*** that L_b is not a regular language.

(a) DFA



-or-

DFA (Q, δ, s, A)

$Q: \{0, 1, 2\}$

$\delta(q, a) = (q + a) \bmod 3$

$s = 0$

$A = \{0\}$

(b) Let $F = \{1^n \mid n \geq 1\}$. $(F = 11^*)$

Let $x \neq y$ be any two distinct strings in F.

$x = 1^i \neq y = 1^j$ for $i \geq 1, j \geq 1, i \neq j$

Let $z = 2^j$.

$xz = 1^i 2^j \in L_6$, because #0s, #1s, #2s are distinct.

$yz = 1^j 2^j \notin L_6$, because $\#(1, yz) = \#(2, yz)$

So z distinguishes $x \neq y$.

So F is an infinite fooling set of L_6

7(a) cont.: Need to prove G has a Ham. cycle iff G' has a Ham bicycle.

⇒ Suppose G has a Ham cycle C.

Make two copies of C, C_0 & C_1, in vertices $(v,0)$ & $(v,1)$.

That's two simple cycles, disjoint, & have every member of V' once.

i.e. $C_0 \cup C_1$ is a Ham. bicycle in G'.

⇐ Suppose G' has a Ham. bicycle $\{C_0, C_1\}$.

No edges between 0 vertices & 1 vertices,

so C_0 is a simple cycle covering all 0 vertices & only 0 vertices.

Remove 0 components from C_0's vertices to get a simple cycle C covering V.

C is a Ham. cycle in G.

5(b) Want a shortest walk from $\{(s,0),(s,G),(s,P)\}$ to $\{(t,0),(t,G),(t,P)\}$.

Use Dijkstra's from each of $\{(s,0),(s,G),(s,P)\}$ & return shortest walk to any of $\{(t,0),(t,G),(t,P)\}$ found.

Time: $O(|E'|\log|V'|) = \underline{O(|E|\log|V|)}$.

6(b): Niknav(i,j,k,ℓ): max score of a Niknav sub-game starting at pos. (i,j) + ending at position (k,ℓ).

Need to compute $\max\{$Niknav$(i,j,k,\ell) \mid 1 \le i \le k \le n,\ 1 \le j \le \ell \le n,\ k = n \vee \ell = n$ (or both)$\}$

$$\text{Niknav}(i,j,k,\ell) = \begin{cases} V[i,j] & \text{if } i = k \text{ and } j = \ell \\ -\infty & \text{if } i > k \text{ or } j > \ell \\ -\infty & \text{if } V[i,j] \neq V[k,\ell] \\ 2 \cdot V[i,j] + \max\{\text{Niknav}(i+1,j,k-1,\ell), \text{Niknav}(i,j+1,k-1,\ell), \text{Niknav}(i+1,j,k,\ell-1), \text{Niknav}(i,j+1,k,\ell-1)\} & \text{o.w.} \end{cases}$$

Memoize in a 4D array.

Eval as for i going down
  for j going down
    for k going up
      for ℓ going up

Time: $O(1) \cdot O(n^4) = \underline{O(n^4)}$

Q(6) (DO NOT GRADE!!!):

In NP: Certificate is assignment of 1, 2, or 3 to each vertex saying which clique it is in.

NP-hard: From 3Color.

Let $G=(V,E)$ be an instance of 3Color.

Build $G'=(V',E')$. $V':=V$

↑ complement $E':=\{u\text{-}v \mid u\text{-}v \notin E\}$

Takes $O(|V|^2)$ (polynomial) time.

Need to prove G has a proper 3-coloring $c: V \to \{1,2,3\}$ iff G' has a clique-partion of size 3.

$\Rightarrow$ Suppose G has a proper 3-coloring $c: V \to \{1,2,3\}$.

Let $V_i = \{v \mid c(v) = i\}$ for $i \in \{1,2,3\}$.

If $u,v \in V_i$ then $u\text{-}v \notin E$, so $u\text{-}v \in E'$.

So V_1, V_2, V_3 is a clique-partition.

$\Leftarrow$ Suppose G' has a clique-partion V_1, V_2, V_3.

Let $c: V \to \{1,2,3\}$ where $c(v) = i$ for all $v \in V_i$ for all i.

If $c(u) = c(v) = i$, then $u\text{-}v \in E'$, so $u\text{-}v \notin E$.

So c is a proper 3-coloring of G.

(overflow / scratch paper)

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?