Homework 5 — Reductions and Approximations

Due: December 8th

A few of these questions may appear on or help you prepare for the quiz on **December 8th**.
Note that these selected solutions and comments do not necessarily represent complete answers.

1. State each of the following problems as decision problems.

   (a) The Minimum Spanning Tree Problem

   Given a connected, weighted graph and a threshold k, determine whether or not there exists a spanning tree of weight $\leq k$.

   (b) The Levenshtein Distance Problem

   Given two strings x and y and a threshold k, determine whether or not it is possible to transform x into y using $\leq k$ deletions, insertions, and substitutions.

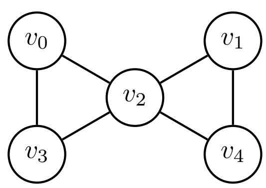
   (c) The Knapsack Problem

   Given a set of items, a weight capacity W, and a threshold k, determine whether or not there exist quantities $x_i \in \{0, 1\}$ such that $\sum x_i \cdot w_i \leq W$ and $\sum x_i \cdot v_i \geq k$.

   (d) The Hamiltonian Path Problem

   Given a connected graph, determine whether or not there exists a path that passes through every vertex exactly once (the Hamiltonian Path Problem is already a decision problem).

2. Given a graph $G = (V, E)$, an *independent set* is a subset of vertices $S \subseteq V$ such that no edges in E connect two vertices in S. For example:



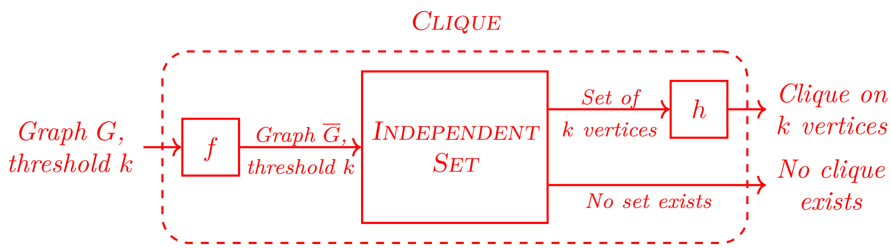
   The (Decision) Independent Set Problem asks for an independent set of cardinality k. In the above graph, $S = \{v_0, v_4\}$ is an independent set, and proof that there exists an independent set of cardinality 2. There also exist independent sets of cardinality 1, but none of cardinality 3, 4, or 5.

   Recall that the (Decision) Clique Problem is known to be $\mathcal{NP}$-Hard. Using this fact, prove that the (Decision) Independent Set Problem is also $\mathcal{NP}$-Hard.

   Consider reducing Clique to Independent Set: note that a clique is completely connected, such that any two vertices in a clique must be adjacent, whereas an independent is completely disconnected, such that any two vertices in an independent set cannot be adjacent.

Proof:

- *Consider reducing (Decision) Clique to (Decision) Independent Set:*



- *Let f be an algorithm that takes as input a graph G and threshold k, and produces as output the complement $\overline{G}$ and the same threshold k.*
- *Let h be an algorithm that takes as input an independent set of k vertices, and produces as output the subgraph induced by that set in G.*
- *Then any vertices in the independent set must not be adjacent in $\overline{G}$, so any vertices in the induced subgraph are adjacent in G, thereby forming a clique. Thus, we have reduced Clique to Independent Set, where f has complexity $O\left(|V|^2\right)$ and h, $O\left(k^2\right)$.*
- *Thus, the (Decision) Independent Set Problem is in $\mathcal{NP}$-Hard.* $\square$

3. Recall that a proposition in *conjunctive normal form* is a conjunction of clauses, where each clause is a disjunction of literals. For example:

$$(p \vee q) \wedge (\neg p \vee r \vee \neg r) \wedge (\neg q) \wedge (s \vee \neg p)$$

The Stingy Satisfiability Problem asks for a satisfying assignment with at most k true variables. In the above proposition, $p \equiv T$, $q \equiv F$, $r \equiv F$, $s \equiv T$ is a satisfying assignment, and proof that there exists one with 2 true variables. There also exists an assignment with 3 true variables, but none with 1 or 4.

Recall that the 3-Satisfiability Problem is known to be $\mathcal{NP}$-Hard. Using this fact, prove that the Stingy Satisfiability Problem is also $\mathcal{NP}$-Hard.

Consider reducing 3-Satisfiability to Stingy Satisfiability: note that any instance of 3-SAT, with the addition of a threshold k, is a valid instance of Stingy SAT. Further note that smaller values of k limit the possible assignments, whereas sufficiently large values of k permit any assignment.

4. Given a family of sets $\mathcal{A}$ in some universe U, a *set cover* is a subset $\mathcal{B} \subseteq \mathcal{A}$ whose union is equal to U. For example:

$$U = \{1, 2, 4, 8, 16\}$$
$$\mathcal{A} = \{\{1, 4\}, \{2, 4, 8\}, \{1, 8\}, \{16\}\}$$

The (Decision) Set Cover Problem asks for a set cover of cardinality k. In the above family, $\mathcal{B} = \{\{1, 4\}, \{2, 4, 8\}, \{16\}\}$ is a subset whose union equals U, and proof that there exists a set cover of cardinality 3. There also exists a set cover of cardinality 4, but none of 1 or 2.

Recall that the (Decision) Vertex Cover Problem is known to be $\mathcal{NP}$-Hard. Using this fact, prove that the (Decision) Set Cover Problem is also $\mathcal{NP}$-Hard.

Consider reducing Vertex Cover to Set Cover: note that both problems involve selecting elements of one set to cover elements of another set. It stands to reason that vertices in V must be represented as sets in $\mathcal{A}$, since they are being selected. Likewise, the edges in E must be represented as elements in U, since they are being covered.

5. Consider the following greedy algorithm for approximating the Independent Set Problem:

ApproximateIndependentSet($G = (V, E)$)

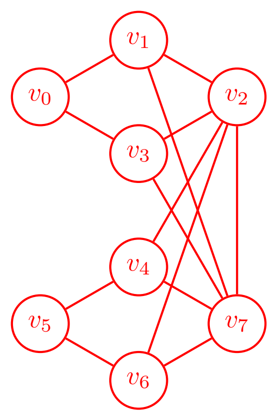
Input: A graph G
Output: An independent set of G

let S be $\emptyset$
while G is not empty **do**
  let v be the vertex of minimum degree in G
  let S be $S \cup \{v\}$
  for all neighbors of v, u **do**
    Remove u from G
  Remove v from G
return S

Given a graph $G = (V, E)$, let $\Delta = \max_{v \in V}\{\deg(v)\}$. Further, let A be the cardinality of the set returned by ApproximateIndependentSet and Opt be that of the maximum independent set in G.

(a) Give an example where $A = $ Opt.

(b) Give an example where $A < $ Opt.



(c) Show that ApproximateIndependentSet is a $(\Delta + 1)$-approximation.

- *Any vertex not in S is a neighbor of at least one vertex in S. For each vertex in S, there exist up to Δ neighbors that cannot be in S: $|V - S| \leq \Delta\,|S|$.*
- *Every vertex is either in S or not in S: $|V| = |V - S| + |S|$, thus, $|V| \leq \Delta\,|S| + |S|$, thus, $|V| \leq (\Delta + 1)\,|S|$. Note that $A = |S|$.*
- *No independent set can be larger than the vertex set: Opt $\leq |V|$, thus, Opt $\leq (\Delta + 1) \cdot A$, thus, ApproximateIndependentSet is a $(\Delta + 1)$-approximation.*

[*Hint:* Note that greedily selecting a vertex v prevents the selection of $\deg(v)$ others.]

6. Consider the following greedy algorithm for approximating the Knapsack Problem:

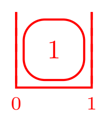
ApproximateKnapsack$(S = \{a_0, a_1, \ldots, a_{n-1}\}, W)$

Input: A finite set S of n items, where $a_i = (w_i, v_i)$, a natural W
Output: A value of a knapsack with capacity W, where items may not be repeated

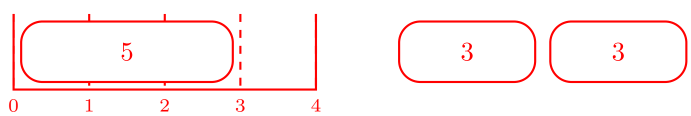
Remove any item $a_i = (w_i, v_i)$ from S if $w_i > W$
let V_A be 0, W_A be W, and A be S, sorted in decreasing order by v_i
for all $a_i = (w_i, v_i) \in A$ **do**
  if $w_i \leq W_A$ **then**
    let V_A be $V_A + v_i$
    let W_A be $W_A - w_i$
let V_B be 0, W_B be W, and B be S, sorted in decreasing order by ${}^{v_i}/_{w_i}$
for all $a_i = (w_i, v_i) \in B$ **do**
  if $w_i \leq W_B$ **then**
    let V_B be $V_B + v_i$
    let W_B be $W_B - w_i$
return $\max\{V_A, V_B\}$

Given a set of items S and a natural W, let A be the value returned by ApproximateKnapsack and Opt be the maximum value of a knapsack with capacity W.

(a) Give an example where $A =$ Opt.



(b) Give an example where $A <$ Opt.



(c) Show that ApproximateKnapsack is a 2-approximation.

- *Let v_j be the first value not included in V_B; if selecting fractions of items were allowed, including the corresponding fraction of v_j in V_B would be optimal: Opt $\leq V_B + v_j$.*
- *No single value can exceed V_A, which always greedily selects the highest value item: $v_j \leq V_A$, thus Opt $\leq V_B + V_A$.*
- *By construction, $A = \max\{V_A, V_B\}$, thus, $A \geq V_A$ and $A \geq V_B$, thus, Opt $\leq A + A$, thus, Opt $\leq 2 \cdot A$, thus, ApproximateKnapsack is a 2-approximation.*

[*Hint:* Let v_j be the first value *not* included in V_B. Note that, if selecting fractions of items were allowed, including the corresponding fraction of v_j in V_B would be optimal. Further note that $v_j \leq V_A$.]