CS 170 HW 11

Due **2021-11-15, at 10:00 pm**

1 Study Group

List the names and SIDs of the members in your study group. If you have no collaborators, you must explicitly write none.

In addition, we would like to share correct student solutions that are well-written with the class after each homework. Are you okay with your correct solutions being used for this purpose? Answer "Yes", "Yes but anonymously", or "No"

2 NP Basics

This is a solo question.

Assume A reduces to B in polynomial time. In each part you will be given a fact about one of the problems. What information can you derive of the other problem given each fact? Each part should be considered independent; i.e., you should not use the fact given in part (a) as part of your analysis of part (b).

1. A is in **P**. $B \in \{P, NP, NPH\}$
2. B is in **P**. A is in P
3. A is **NP**-hard. B is NPH
4. B is **NP**-hard. $A \in \{NPH, NP, P\}$

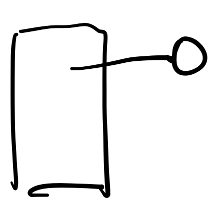
3 Runtime of NP

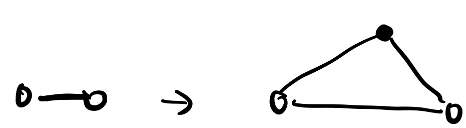
True or False (with brief justification): Suppose we can show for some fixed k, an NP-complete problem P has a time $O(n^k)$ algorithm. Then every problem in NP has a $O(n^k)$ time algorithm. ✓

4 Dominating Set

A dominating set of a graph $G = (V, E)$ is a subset D of V, such that every vertex not in D is a neighbor of at least one vertex in D. Let the Minimum Dominating Set problem be the task of determining whether there is a dominating set of size $\leq k$. Show that the Minimum Dominating Set problem is NP-Complete. You may assume that G is connected.

Vertex cover





5 More Reductions

Given an array $A = [a_1, a_2, \dots, a_n]$ of nonnegative integers, consider the following problems:

1 **Partition**: Determine whether there is a subset $P \subseteq [n]$ ($[n] := \{1, 2, \cdots, n\}$) such that $\sum_{i \in P} a_i = \sum_{j \in [n] \setminus P} a_j$

2 **Subset Sum**: Given some integer k, determine whether there is a subset $P \subseteq [n]$ such that $\sum_{i \in P} a_i = k$

3 **Knapsack**: Given some set of items each with weight w_i and value v_i, and fixed numbers W and V, determine whether there is some subset $P \subseteq [n]$ such that $\sum_{i \in P} w_i \leq W$ and $\sum_{i \in P} v_i \geq V$

For each of the following clearly describe your reduction, justify runtime and correctness.

(a) Find a linear time reduction from SUBSET SUM to PARTITION.

$\frac{\text{Sum } a}{2} = \sum_{i \in P} a_i$

(b) Find a linear time reduction from SUBSET SUM to KNAPSACK.

6 Orthogonal Vectors

In the 3-SAT problem, we have n variables and m clauses, where each clause is the OR of (at most) three of these variables or their negations. The goal of the problem is to find an assignment of variables that satisfies all the clauses, or correctly declare that none exists.

In the orthogonal vectors problem, we have two sets of vectors A, B. All vectors are in $\{0, 1\}^m$, and $|A| = |B| = n$. The goal of the problem is to find two vectors $a \in A, b \in B$ whose dot product is 0, or correctly declare that none exists. The brute-force solution to this problem takes $O(n^2 m)$ time: We compute all $|A||B| = n^2$ dot products between two vectors in A, B, and each dot product takes $O(m)$ time.

Show that if there is a $O(n^c m)$-time algorithm for the orthogonal vectors problem for some $c \in [1, 2)$, then there is a $O(2^{cn/2} m)$-time algorithm for the 3-SAT problem. For simplicity, you may assume in 3-SAT that the number of variables must be even. *Hint: Try splitting the variables in the 3-SAT problem into two groups.*

m

truth : 0 false : 1

n

$|B| = n$

$(2^{\frac{n}{2}})^c m$ $|A| = \frac{n}{2}$ $\frac{n}{2} = |B|$

$(x_1, x_2, x_{...}, x_n$

$$0, 1 \qquad 2 \big] = \frac{n}{2}$$

$$\overbrace{(x_1 \vee x_2 \vee x_3)}^{m_1} \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_3)$$

3 variables 2 clauses

$(x_1 \vee x_2 \vee x_3) \wedge$

$(x_1 \vee \bar{x}_2)$

$(\bar{x}_1 \vee \bar{x}_2)$

$(\bar{x}_4 \vee$

$$\overset{A}{[\ \ ]} \cdot \overset{B}{[\ \ ]} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}$$

$2^{\frac{n}{2}} \qquad 2$

F F

$x_2 \ x_4 \qquad x_3 \ x_0$

x_1 in A $\qquad$ ~~x_4~~ x_3 in B

$x_1 . x_2$ in A $\qquad x_3 . x_0$ in B

R	x_1	x_2	x_3	x_4				
7	0	0	0	0	0	1	0	0
	0	0	0	1	0	1	0	1
	0	0	1	0	0	1	1	0
	0	0	1	1	0	1	1	1

$$\begin{array}{rccc} & x_1 \vee x_2 & & x_3 & & \\ \text{first clause} & \begin{bmatrix} 0 \\ 1 \end{bmatrix} & \cdot & \begin{bmatrix} 0 \\ 0 \end{bmatrix} & = & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \text{second clase} & & & & & \\ & \bar{x}_1 \vee \bar{x}_2 & & x_4 & & \end{array}$$

1 Study Group

none YES

2 NP Basics

If A reduces to B, we know B can be used to solve A, which means B is at least as hard as A.

(a) B can be anything {P,NP,NPH}

(b) A is in P

(c) B in NP-hard

(d) A can be anything {P,NP,NPH}

3 Runtime of NP

False

By definition, NP problem can be reduced to NPC in polynomial time

If all NP problem have a O(n^k) runtime, then we have that any NP problem can be reduced to the NPH problem in linear time. But we only have NP can be reduced to NPC in polynomial time. Contradiction!

Thus, every problem in NP doesn't guarantee tohave a O(n^k) time algorithm

4 Dominating Set

We can prove Dominating Set as NPC by showing that the vertex cover can be reduced by dominating set

Given a graph G(V,E) and a number k, we define Vertex Cover = {<G,k>: there exists a vertex cover of G with size at most k}

For (u,v) in E, we add a new vertex a and new edges (u,a) and (a,v) to G. Denote the new graph G'(V',E')

1)If G' has a dominating set of size k

Notice that we create a triangle for u,v and a if (u,v) is in E. By definition, at least one of the 3 vertex belongs to D. The new vertex we added only adjacent to u and v. Thus, if the dominating set contains a, we can switch that vertex with u or v. Also notice that if we have u or v in the dominating set, there is no need to add a into the set. Thus, we can assume that there is no a in the dominating set without making the set size larger. Thus, the dominating set contains u or v for (u,v) in E. So it's a vertex cover in G.

2)If G has a vertex cover of size at most k

It's clear that the vertex cover for G is the dominating set for G'. Because for every edge (u,v) in E, u or v belongs to D and a is connected to u and v.

5 More Reductions

(a)Let (L,t) be an instance of subset's sum where L holds a subset of A and B is the target sum.

Let m be the sum of elements in L. Let L' be the set L + α(n+1) + α(n+2) where α(n+1) = 2m + t, α(n+2) = 3m-t.

Then L' is a partition iff (L,t) is a subset sum.

If (L,t) is a subset sum, there exist B which is a subset of L with sum t. Then (L-B)U{α(n+1)} and BU{α(n+2)} is a partition

If L' is a partition, there exists a partition (*L1*,*L2*) . Since the total sum is 6m, α(n+1) and α(n+2) should not appear in the same set. WLOG, let α(n+1) in *L1* and α(n+2) in *L2*. Then *L2* - {α(n+2)} has sum k

Notice that we L' can be constructed in linear time. Thus, the reduction is linear

(b)Set w_i = v_i = a_i i in {1,2,...,n} and W = V = t which is the target sum.

Then P is the subset of the modified knapsack iff P is a subset sum with t. The reduction is linear

If P is the subset of knapsack, then we know that the sum of elements in P is less or equal then t from weight constraint and the sum of elements in P is greater or equal then t from value constraint. Thus, the sum of elements in P is t which is a subset sum with t.

If P is the subset sum with t, then P is also the subset of knapsack problem since k<=k , k>=k

6 Orthogonal Vectors

First, let 0 represents true and 1 represents false

For a 3-SAT problem with n variables and m clauses, we divide n variables into 2 parts A = {x_1,..,x_n/2}

B = {x_n/2+1,...,x_n}. There are 2^(n/2) permutations for A and B respectively.

Let's discuss about a specific permutation. Suppose we have a bit string of length n.

Then construct the dot product [ ]*[ ] (Denote the 2 bracket to be A' and B'). Each multiplier has m rows. We want the result to be a zero vector to make it satisfiable.

Suppose the i-th clause looks like (x_pVx_qVx_r).

If x_p, x_q and x_r are all in A', then the i-th entry for A' is the result of x_pVx_qVx_r while i-th entry of B' is 0

If one of them has 2 variables, WLOG, let's say x_p and x_q belongs to A' and x_r belongs to B'. The i-th entry of A' is the result of x_pVx_q while the i-th entry of B' is x_r.

Construct all i in {1,2,...,m}. Compute the product and check whether it's a zero vector

Notice that we assume all vectors in A, B are in {0,1}^m and |A| = |B| = n with O(n^c*m)-time to find the orthogonal product.

Since we assume the 3-SAT problem has n variables and m clauses, we need 2^n/2 permutations. Thus, the 3-SAT takes time O((2^n/2)^c*m)