CS-3510-C F23 Exam 4 Version B

Vidit Dharmendra Pokharna

TOTAL POINTS

87 / 110

QUESTION 1

Question 1 30 pts

1.1 (i) **3 / 3**

✓ **- 0 pts** *Correct*

- 3 pts Incorrect

1.2 (ii) **3 / 3**

✓ **- 0 pts** *Correct*

- 3 pts Incorrect

1.3 (iii) **0 / 3**

- 0 pts Correct

✓ **- 3 pts** *Incorrect*

1.4 (iv) **3 / 3**

✓ **- 0 pts** *Correct*

- 3 pts Incorrect

1.5 (v) **3 / 3**

✓ **- 0 pts** *Correct*

- 3 pts Incorrect

1.6 (vi) **0 / 3**

- 0 pts Correct

✓ **- 3 pts** *Incorrect*

1.7 (vii) **3 / 3**

✓ **- 0 pts** *Correct*

- 3 pts Incorrect

1.8 (viii) **0 / 3**

- 0 pts Correct

✓ **- 3 pts** *Incorrect*

1.9 (ix) **3 / 3**

✓ **- 0 pts** *Correct*

- 0 pts Incorrect

1.10 (x) **3 / 3**

✓ **- 0 pts** *Correct*

- 3 pts Incorrect

QUESTION 2

2 Question 2 10 / 10

✓ **- 0 pts** *Correct*

- 3 pts Minor Error

- 7 pts ``Major Error (ex. More than 2 edge cases exist)

- 8 pts Completely Incorrect (Proved NP-C)/ Non poly-time algorithm provided

- 10 pts Missing

QUESTION 3

3 Question 3 10 / 10

✓ **- 0 pts** *Correct*

- 3 pts Minor Error

- 7 pts ``Major Error (ex. More than 2 edge cases exist)

- 8 pts Completely Incorrect (Proved NP-C)/ Non poly-time algorithm provided

- 10 pts Missing

QUESTION 4

Question 4 25 pts

4.1 (i) **5 / 5**

✓ **- 0 pts** *Correct*

- 5 pts Incorrect/ Missing NP proof

- 3 pts Major Error in NP Proof

- 2 pts Minor Error in NP Proof

4.2 (ii) **15 / 20**

- 0 pts Correct

- 20 pts Missing

- 17 pts Completely Incorrect / Reverse reduction provided (k-kite $\rightarrow$ NP-complete problem)

- 8 pts Major Error in Reduction (ex. reduction cannot be justified)

✓ **- 5 pts** *Minor Mistake in Reduction (ex. small mistake in converting clique to k-kite)*

- 2.5 pts Incorrect/ Missing $x \in A \implies f(x) \in B$

- 2.5 pts Incorrect/ Missing $x \notin A \implies f(x) \notin B$

Need to specify that you have to add a tail to every single vertex of the graph to ensure that there is a kite.

QUESTION 5

Question 5 25 pts

5.1 (i) **5 / 5**

✓ **- 0 pts** *Correct*

- 5 pts Incorrect/ Missing NP proof

- 3 pts Major Error in NP Proof

- 2 pts Minor Error in NP Proof

Iterating through the vertices and edges takes O(V+E)

5.2 (ii) **20 / 20**

✓ **- 0 pts** *Correct*

- 0.5 pts Reduction augments graph needlessly. Can simply set n = 3. Additionally, the value of n isn't specified in the input.

- 20 pts Missing

- 17 pts Completely Incorrect / Reverse reduction provided (N-coloring $\rightarrow$ NP-complete problem)

- 8 pts Major Error in Reduction (ex. reduction cannot be justified)

- 5 pts Minor Mistake in Reduction (ex. small mistake in converting 3-coloring to n-coloring)

- 2.5 pts Incorrect/ Missing $x \in A \implies f(x) \in B$

- 2.5 pts Incorrect/ Missing $x \notin A \implies f(x) \notin B$

QUESTION 6

6 Question 6 1 / 10

- 0 pts Correct

- 10 pts Missing/Completely Incorrect

✓ **- 9 pts** *Major Error / Reverse reduction provided (Clique-cover $\rightarrow$ NP-complete problem)*

- 5 pts Minor Mistake in Reduction (ex. small mistake in converting 3-coloring to clique-cover)

- 2.5 pts Incorrect/ Missing $$x \in A \implies f(x) \in B$$

- 2.5 pts Incorrect/ Missing $$x \notin A \implies f(x) \notin B$$

This approach doesn't work. If there isn't a vertex cover of some size x, there could still be k cliques in the clique cover with sizes other than x.

GTID: 903772087
NAME: Vidit Pokharna

Georgia Institute of Technology

Fall 2023

CS 3510C – Design & Analysis of Algorithms
Exam 4 (Version B)

November 16, 2023

TIME ALLOWED: 75 MINS

Name: Vidit Pokharna

GTID: 903772087

GT Username: vpokharna3

INSTRUCTIONS TO CANDIDATES

1. Please write your NAME and GTID clearly on all the pages.
2. This examination paper is worth **110 points** (100 + 10 extra credit points) comprising of **SIX (6)** questions and ELEVEN printed pages.
3. **ONLY** write on the front sheets of paper that are numbered. The backs will not be scanned.
4. Wherever a blank is provided, please write your answer in the given blank.
5. Calculators are **NOT** allowed.

I am in aware of and the accordance with Academic Honor Code of Georgia Tech and the Georgia Tech Code of Conduct. I'll use no external help on this test. Also, I have read all the instructions on this page.

Signature: Vidit

GTID: 903772087
NAME: Vidit

You can **only** use the following NP-Complete problems for the reductions in this exam:

1. `SAT`
2. `3-SAT`
3. `Independent Set`
4. `Clique`
5. `Vertex Cover`
6. `Knapsack`
7. `Subset Sum`
8. `3-Coloring`

You may also use `2-SAT` as a black box.

Note: Feel free to use the rest of this page to extend your answer to any questions or for rough work. In case you're using it for answer extension, please make sure to make a reference.

GTID: 903772087
NAME: Vidit

Problem 1 *(30 points; 3 points each)*

Indicate whether the following statements are ***true*** or ***false***.

(i) Suppose $A \leq_p B$, and $C \leq_p B$. If we have that $C \leq_p A$, then $B \leq_p A$.

Answer: False

(ii) Every NP-Complete problem is NP-Hard.

Answer: True

(iii) If the best algorithm to solve 3-SAT takes exponential time, then there must exist a problem in NP which is neither NP-Complete nor in P.

Answer: False

(iv) Suppose you choose any NP-hard problem F. It is necessary that you can verify a candidate solution S to F in polynomial time.

Answer: False

(v) There exists a reduction from 2-SAT to Super Mario Brothers.

true *Answer:* True

(vi) Every problem in NP is solvable in exponential time.

Answer: False

(vii) Suppose problem A is in P. If you reduce another problem B to A using a polynomial-time transformation, B must be in P.

Answer: True

GTID: 903772087
NAME: Vidit

(viii) Consider the problem `5-Size-Clique` which takes in a graph $G = (V, E)$ and outputs whether the graph contains a clique of size 5. `5-Size-Clique` is NP-Complete.

Answer: True

CLIQUE

$k = 5$

(ix) Consider

$$f(x_1, x_2, x_3, x_4, y) = (x_1 \vee x_2 \vee y) \wedge (\overline{y} \vee x_3 \vee x_4)$$

F F T F F F

and

$$g(x_1, x_2, x_3, x_4, y, z) = (x_1 \vee x_3 \vee z) \wedge (\overline{z} \vee x_2 \vee y) \wedge (\overline{y} \vee x_4)$$

F T F F T F

f and g will have identical results for all assignments x_1, x_2, x_3, x_4 regardless of y and z.

Answer: True

$x_1 \vee x_3 \vee x_2 \vee y$

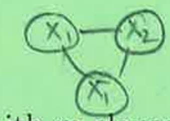
(x) If a polynomial time algorithm is discovered to solve `Vertex Cover`, then there exists a polynomial-time algorithm for `3-SAT`.

Answer: True

GTID: 903772087
NAME: Vidit

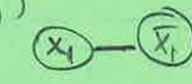
Problem 2 *(10 points)* $(x_1 \land x_2 \land x_3) \lor (\bar{x}_1$

Show that the following DNF-SAT problem is in class P.



Input: A boolean formula in Disjunctive Normal Form with m clauses and n variables.

$\neg((x_1 \lor x_2 \lor \bar{x}_1) \land (x_3 \lor x_2 \lor \bar{x}_2) \land (x_3 \lor x_2 \lor \bar{x}_1))$



Example Input: $f(x) = (x_1 \land x_2 \land \overline{x}_1) \lor (x_3 \land x_2 \land \overline{x}_2) \lor (x_3 \land x_2 \land \overline{x}_1)$

$\bar{x}_1 \lor \bar{x}_2 \lor x_1$

Output: Determines whether there exists a satisfying assignment such that the boolean formula evaluates to True.

To solve this in polytime, we have to iterate through the clauses and see if there exists any contradictions (eg: x_1 and $\bar{x}_1$). If such exists, that clause will be false. However, if no contradictions exist then the assignment can be created in such a way so it evaluates to true. All we need is one true clause, so we can stop there, and we can say there exists a satisfying assignment.

This will take $O(mn)$ time, which is polytime.

DNF-SAT $\in$ P

GTID: 903772087
NAME: Nidit

Problem 3 *(10 points)*

Show that the following Negative-4SAT problem is in class P.

Input: A boolean formula in CNF with m clauses and n variables where each clause contains at most 4 variables. Additionally, all clauses only contain negated variables $\overline{x_i}$. That is, literal x_i does not exist in any of the m clauses.

$(\overline{x_1} \vee \overline{x_2} \vee \overline{x_3} \vee \overline{x_4}) \wedge (\overline{x_2} \vee$

Output: Determines whether there exists a satisfying assignment such that the boolean formula evaluates to True.

There will always be an existing ~~sat~~ satisfying assignment because if you set each ~~literal~~ to false, the entire formula will evaluate to true since each negation makes the variable true, and will all true variables, all clauses are true. Thus the assignment of all literals x_i being false would satisfy this problem.

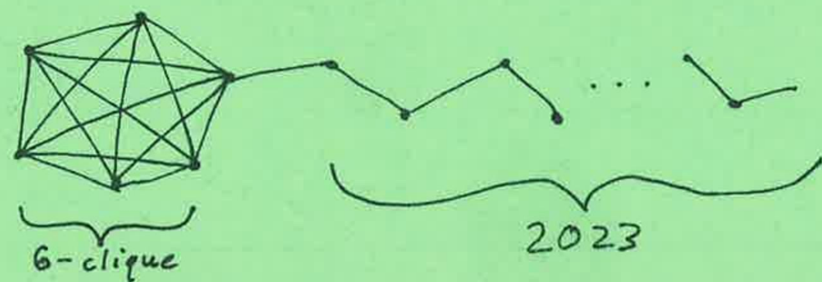
Setting each ~~literal~~ to false would take $O(n)$ time, which is polytime.

Negative-4SAT $\in$ P

GTID: 903772087
NAME: Vidit

Problem 4 *(25 points)*

A kite is a graph on $k + 2023$ vertices, in which k of the vertices form a clique and the remaining 2023 vertices are connected in a "tail" that consists of a path joined to one of the vertices of the clique. Show that the following `k-KITE` problem is NP-Complete.



Input: Graph $G = (V, E)$ with a natural number $k \geq 3$.

Output: Determines whether there exists a kite on $k + 2023$ vertices in G.

(i) Show the problem is in class NP.

We can check the clique and tail individually.
Checking the tail takes $O(2023)$ time, ensuring the format of it is proper (as a tail). We can further check the clique by seeing if each vertex in the clique is connected to every other vertex by a unique edge. This takes $O(k^2)$ time. then, we should check if the clique and tail are connected, which is constant time.

Thus, we can verify a witness solution in polytime.

k-kite $\in$ NP.

GTID: 903772087
NAME: Vidit

(ii) Show the problem is in class NP-Hard.

We can use the NP-complete problem CLIQUE as reference for this. Given a CLIQUE problem with graph G and integer k, we can convert it into k-kite problem $G' = G + (\text{a tail})$, $k' = k$, $V' = k + 2023$, $E' = \frac{k(k-1)}{2} + 2023$. Since we are simply adding a tail of known length, it is a direct mapping plus a constant addition to the graph, which is a polytime reduction.

$\Phi \in$ CLIQUE $\rightarrow f(\Phi) \in$ k-kite: Given a solution to CLIQUE problem, our transformation would create a kite on k+2023 vertices. Giving the mapping of our transformation, G' would contain a clique of k vertices with the additional tail of 2023 vertices, which satisfies k-kite.

$\Phi \notin$ CLIQUE $\rightarrow f(\Phi) \notin$ k-kite: If we are given Φ that isn't a clique, then our transformation of adding a tail would not fully satisfy k-kite. Since we only have the 2023 vertices tail and not the clique, this would not satisfy k-kite.

Thus, k-kite $\in$ NP-Hard since we showed a polytime reduction from an NP-complete problem that maps properly

Therefore, since k-kite $\in$ NP and k-kite $\in$ NP-Hard, k-kite is NP-complete.

GTID: 903772087
NAME: Vidit

Problem 5 *(25 points)*

Show that the following `N-Coloring` problem is NP-Complete.

Input: A graph $G = (V, E)$ and $n \in \mathbb{N}$

Output: Determines whether there exists a coloring of the graph using n colors. Recall that a coloring of a graph requires that for every edge (u, v), vertices u and v must not have the same color.

(i) Show the problem is in class NP.

Given a witness (colored) graph, we have to ensure the graph is colored with ~~less than or equal to~~ n colors ~~(since if less colors are used, it is still possible to color with n colors)~~ and that no two adjacent vertices have the same color. We can use BFS to do this, checking adjacent vertices, and counting the number of colors used, placing them in a set once a new color is found. This would take $O(n(|V| + |E|))$ time as we check color set while running BFS, which is polytime.

Thus, N-coloring $\in$ NP.

GTID: 903772087
NAME: Vidit

(ii) Show the problem is in class NP-Hard.

We can use the NP-complete problem 3-coloring as reference for this reduction.

Given a 3 coloring problem as graph G, we can transform it into an N-coloring problem where G' = G and n = 3. Since this is a direct mapping, it will take polytime to set up the graph (constant).

$\phi \in$ 3coloring $\rightarrow f(\phi) \in$ N-coloring: If ϕ is a satisfiable 3-coloring, our transformation would allow us to have a graph connected such that we would have to use 3 colors since they're all connected. This fits the n-coloring scheme and thus $f(\phi)$ would be a satisfiable N-coloring solution, given the transformation with n = 3.

$\phi \notin$ 3coloring $\rightarrow f(\phi) \notin$ N-coloring: If ϕ is not a satisfiable 3 coloring, then our transformation would give a graph that would require more than n = 3 colors, making it unsatisfiable for N-coloring with the current transformation.

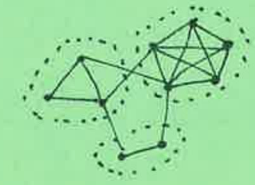
Thus, N-coloring $\in$ NP-Hard, and since N-coloring $\in$ NP (3coloring $\leq_p$ N-coloring) and N-coloring $\in$ NP-Hard

N coloring is NP-complete.

GTID: 903772087
NAME: Vidit

Problem 6 *(Extra Credit; 10 points)*

A clique-cover is a partition of the vertices V of a graph into sets $S_1, S_2, \ldots, S_k$ such that the vertices in each set S_i form a clique of size 2 or greater. Show that the following `Clique-Cover` problem is NP-Hard.



Clique-cover with 3 cliques.

Input: Graph $G = (V, E)$ with a natural number $k \geq 2$.

Output: Determines whether there exists a clique-cover of size k. That is, you need to find whether G can be partitioned into k cliques.

$|V|$ if $|V|$ is even
$|V|+1$ if $|V|$ is odd

We can use vertex cover as reference.
Given graph G, we can find a vertex cover set of size $|V|$ and transform it into Clique-cover problem with $G' = G$. and $k = |V|$ or $|V|+1$ dep on parity
This takes polytime to transform.

$\phi \in$ vertex cover $\rightarrow f(\phi) \in$ Clique-cover: If ϕ is a satisfiable vertex cover, then we can make a clique cover with cliques of size 2 for them all, partitioning into $\lceil \frac{|V|}{2} \rceil$ cliques. This is satisfiable for clique-cover.

$\phi \notin$ vertex cover $\rightarrow f(\phi) \notin$ clique cover: If ϕ is not in vertex cover, then we cannot necessarily make a clique cover as there is no guarantee of a connected graph, this is not satisfiable for clique cover.

Thus, Clique-cover $\in$ NP-Hard.