CS180 HW8

Siddharth Joshi

TOTAL POINTS

100 / 100

QUESTION 1

1 Problem 1 **25 / 25**

- ✓ **- 0 pts** Correct
- **- 15 pts** failed to construct the bipartitie graph G correctly
- **- 10 pts** failed to use Hall's theorem
- **- 20 pts** wrong answer but showed efforts
- **- 25 pts** wrong answer or no answer
- **- 0 pts** Click here to replace this description.

QUESTION 2

2 Problem 2 **25 / 25**

- ✓ **- 0 pts** Correct
- **- 15 pts** failed to construct G'
- **- 20 pts** wrong answer but showed efforts
- **- 25 pts** wrong answer or no answer

QUESTION 3

3 Problem 3 **25 / 25**

- ✓ **- 0 pts** Correct
- **- 25 pts** No answer found

QUESTION 4

4 Problem 4 **25 / 25**

- ✓ **- 0 pts** Correct
- **- 25 pts** No answer found

1: Round Robin Tournament (Hall's Theorem)

Proof. Showing that the idea of a round robin tournament doesn't violate Hall's condition:
$|N(X)| \geq |X|$ for every non-empty subset X of the smaller set of vertices in the bipartite graph
The smaller set of vertices is obviously the set of days as there are 2n - 1 days and 2n teams.
Let D be the set of vertices representing the 2n - 1 days of the tournament.
Let T be the set of vertices representing the 2n teams of the tournament.
The question now in formal graph theory becomes does there exist a matching of size 2n - 1 in the bipartite graph with vertices D ∪ T with every vertex t ∈ T has an edge to a vertex d ∈ D if and only if team t won on day d.
Arguing that Hall's condition is not violated by any set X ⊆ D (proof by contradiction)
Assume to the contrary that there is a set X ⊆ D such that the neighborhood of X i.e. N(X) is such that $|N(X)| \ngeq |X|$.
N(X) however represents the set of teams that each won on some day d ∈ X. Therefore N(d) ⊆ X.
But since all teams play every day there must be at least n winners every day and hence $|N(d)| \geq$ n and hence $|N(X)| \geq$ n.
Hence in the case that $|X| \leq$ n, it is easy to see that our initial assumption leads to a contradiction.

Thus now considering the case where $|X| >$ n. But if $|X| >$ n, the same n teams couldn't have won on all days ∈ X as this would need them to play some teams more than once. Therefore for $|X| = n + 1$, $|N(X) \geq |X|$. (Let this be the base case of the induction)
Hence assume for $|X| = n + k$, $|N(X)| \geq n + k$
But $|N(X)| = n + k$ or $|N(X)| > n + k$
However, if $|N(X)| > n + k$ then if a new day d is added to the set X to make it have cardinality n + k + 1, the relation $|N(X)| \geq n + k + 1 = |X|$ is trivially true.
If $|N(X)| = n + k$ then when a new day d is added to the set X to make it have cardinality = n + k + 1, then by the pigeon hole principle this implies that if the neighborhood were to remain to be of size n + k, some teams would have played more than once.
However, this is a contradiction.

This concludes our proof and therefore, Hall's condition is true for all X ⊆ D, therefore there exists a matching of size 2n - 1 which implies by definition that it is possible to pick one winning team everyday without repeating any team.

1 Problem 1 **25 / 25**

- ✓ **- 0 pts** Correct
- **- 15 pts** failed to construct the bipartitie graph G correctly
- **- 10 pts** failed to use Hall’s theorem
- **- 20 pts** wrong answer but showed efforts
- **- 25 pts** wrong answer or no answer
- **- 0 pts** Click here to replace this description.

2: Proving NP-Completeness of Independent Set Problem using Clique

Algorithm:
The main idea is to create an algorithmic box for Clique using only polynomial time to convert the original input G into a graph G' as input for a black box for Independent-Set s.t. the answer to whether there is a clique of size at least k in G is yes if and only if the answer to whether there is an independent set of size at least k in G' is yes. To do this, the input in G must be changed s.t. by finding the largest independent set in G' we instantly know now the largest clique in G. This can be done by taking the complement graph $\overline{G}$ the graph where every pair of vertices u, v have an edge between them i.e (u, v) $\in$ E' if and only if (u, v) $\in$ in E and passing this as the input to G', while setting k' = k.

Running time:
It is easy to see that the runtime of the reduction is polynomial as the complement of a graph can definitely be obtained in at most O($|V|^2$) time and k' is simply directly set to k.

Correctness:
This is more interesting as to prove that this reduction is correct we must now prove:

A clique of size $\geq$ k in G $\iff$ An independent set of size $\geq$ k' in G'

Proving first that: An independent set of size $\geq$ k' in G' $\implies$ A clique of size $\geq$ k in G
A set of independent vertices in G' is by definition a set of vertices where no edge e $\in$ E' is adjacent to two or more vertices in the set. This makes obvious the fact that no two vertices in the independent set of G having an edge connecting them. But G' = $\overline{G}$ and hence every pair of vertices that do not have an edge connecting them in G' must be connected by an edge in the original input graph G. Therefore, since no two vertices in the independent set found in G' have an edge connecting them, the same set of vertices in G is a clique as there must exist an edge between every possible pair of vertices within the set. Hence we have now shown that the set of vertices in the independent set of G' of size $\geq$ k form a clique in G of same cardinality i.e. some value $\geq$ k. QED

Now proving the implication in the other direction i.e.: A clique of size $\geq$ k in G $\implies$ An independent set of size $\geq$ k' in G'
A clique in G by definition has an edge between every two vertices and hence in $\overline{G}$ no two vertices in this set have an edge connecting them. But such a set of vertices in G' is by definition an independent set and its cardinality must be same as that of the clique in G i.e. if the clique in G has size $\geq$ k so does the independent set in G'. QED

Hence we can conclude that the Independent Set Problem is NP-Complete as well.

2 Problem 2 **25 / 25**

- ✓ **- 0 pts** Correct
- **- 15 pts** failed to construct G'
- **- 20 pts** wrong answer but showed efforts
- **- 25 pts** wrong answer or no answer

3: Showing Set Cover, Hitting Set and Dominating Set are Poly-Time Reducible to Each Other

(1) Reducing Set Cover to Hitting Set

Algorithm:

1: Let every element in the original problem for Set Cover be a set in the Hitting Set
2: Let every set in the original problem be an element in the Hitting Set problem
3: Let every set in the new problem have elements as the sets the elements in the original problem used to belong to i.e. if element a $\in$ S then element S $\in$ set a in the new problem for Hitting Set

Running time:
Assuming there are n elements and m sets, the time complexity for creating this new input set is O(mn) which is polynomial and hence this is a valid poly-time reduction.

Correctness:
Creating a sample input for Hitting Set using the input to Set Cover:
Assume, 3 sets s.t. A = a, b, c; B = b, d; C = c, d
This results in the revised input having 4 sets:
a = A
b = A, B
c = A, C
d = C

A Set Cover of size k $\iff$ A Hitting Set of size k'=k with sets and elements reversed as specified

Prove that a Set Cover of size $\leq$ k in the original problem $\implies$ a Hitting Set of size $\leq$ k' = k in the revised input
A Set Cover of size $\leq$ k in the original problem implies that it is possible to select $\leq k$ sets to guarantee that all elements belong to at least one of the sets. But the Hitting Set in the new problem implies that every element in the original problem belongs to at least one set in the original problem due to the inversion of sets and elements. These statements are equal, hence proven.
Prove that a Hitting Set of size $\leq$ k' = k in the revised input $\implies$ a Set Cover of size $\leq$ k in the original problem
This argument is identical to the previous one and is evidently true due to the construction of the revised input. QED

(2) Reducing Hitting Set to Dominating Set

Algorithm:

1: Let every element in the sample space provided to Hitting Set be a vertex in G
2: Let two vertices u and v $\in$ G have an edge connecting them $\iff$ the elements they correspond to are in the same set in the original input
3: Run Dominating Set on G asking the question whether there is a Dominating Set of size $\leq$ k

Running time:

Constructing the graph in the manner specified takes $O(n^2)$ time in the worst case and hence this is a valid poly-time reduction.

Correctness:

Proof by contradiction for: There exists a Dominating Set in G of size $\leq k \implies$ there exists a Hitting Set of $\leq k$ elements.
Assume to the contrary that the dominating set in G does not represent the set of elements that need to be selected to obtain a Hitting Set in the original problem.
However this implies that there is a set S that belongs to the original problem such that none of the elements in the Dominating Set found belong to it. This implies that there is a vertex in G (by construction) that is not part of the Dominating Set nor is it adjacent to any vertex in the Dominating Set. But the property of a Dominating Set guarantees that every vertex v in V either $\in$ the Dominating Set or adjacent to some vertex u s.t. u $\in$ Dominating Set. However this is a contradiction, therefore, the existence of a Dominating Set $\leq k \implies$ the existence a Hitting Set of $\leq k$ elements.

Proof for: there exists a Hitting Set $\leq k \implies$ a Dominating Set of size $\leq k$ in G.
A Hitting Set in the original problem is represented by a set of vertices X in G. However the Hitting Set guarantees that there is at least one element in every set, and since all the elements of a set are adjacent to each other, having just one element for each set guarantees that every vertex belonging to any set has been *dominated* by selecting the vertices in the Hitting Set and therefore all the vertices in G have been dominated and hence the Hitting Set in the original problem represents a Dominating Set in G.

Hence there exists a Hitting Set $\leq k \iff$ there exists a Dominating Set of size $\leq k$ in G. QED

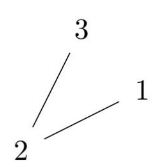
(3) Reducing the Dominating Set problem to Set Cover

Algorithm:

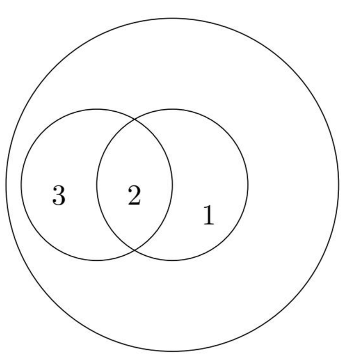
1: Let k' = k
2: Let every vertex v $\in$ G be a Set for the Set Cover Problem input
3: Let the elements $\in$ the set representing vertex v be elements representing the vertices adjacent to it and itself (i.e. the vertices it could potentially dominate)
4: Run Set Cover on the input that was constructed by this process asking the question whether there is a collection of sets $\leq$ k' s.t. their union contains every element in the union of all the sets

Running time:
Instantiating the elements using vertices takes $O(|V|)$ time and traversing all edges to add adjacent vertices to the sets appropriately takes $O(|E|)$ time. Therefore overall time complexity of the reduction is $O(|V| + |E|)$ therefore the reduction in poly-time.

Correctness:
The Set Cover problem generated is such that there is both an element and a set representing every vertex such that every set version of a vertex has its the element versions of its neighbors as its elements. Consider the following example, the input graph G to Dominating Set as such:



would be converted to create input for the Set Cover problem using the reduction in the manner where the outermost circle represents the set corresponding to vertex 2 and hence has itself and all it's neighbors, while the smaller intersecting circles inside represent the sets corresponding to and contain themselves and 2 in the intersection:



Proving that there exists a Set Cover of size $\leq$ k in the new input $\iff$ there exists a Dominating Set of size $\leq k$ in G:
The way the new input has been constructed, a selection of Sets to form a set cover that by definition contain all the elements of the sample space is equivalent to a selection of set of vertices

such that all vertices $\in$ V are either in the set or adjacent to a vertex in the set (as elements of all the sets are neighbors of the corresponding vertex). Since this proves a stronger relation that there exists a Set Cover of size s in the new input $\iff$ Domination Set of size s in the original graph G, the original statement is also proven true.

3 Problem 3 **25 / 25**

✓ **- 0 pts** Correct

- 25 pts No answer found

4: Showing that Hamiltonian Cycle and Hamiltonian Path are Poly-Time Reducible to Each Other

(1) Reducing Hamiltonian Cycle to Hamiltonian Path

Algorithm:
It must be proved rigorously that there exists a hamiltonian cycle in G if and only if there exists a hamiltonian path and an edge connecting the start and end of the path (can be showing using the understanding that a cycle doesn't have any specified starting point/ending point, so without loss of generality any 2 vertices can be assumed to the 'starting' and 'ending' points of the underlying path in a colloquial sense. Thus formalizing this notion:

for edge e = (u, v) $\in$ E **do**
    Let G' $\Leftarrow$ G
    Let u' $\Leftarrow$ a new vertex in G' s.t. it is adjacent only to u
    Let v' $\Leftarrow$ a new vertex in G' s.t. it is adjacent only to v
    if there exists an Hamiltonian Path P in G' (using blackbox for Hamiltonian Path) **then**
        Return True
Return False

Running time:
The for loop runs once for for every edge, therefore time complexity is $O(|E|)$ which is polynomial in input size, hence this reduction is poly-time.

Correctness:
Proving that there exists a Hamiltonian Path P in a G' $\implies$ *there exists a Hamiltonian Cycle C in G.* But this is equivalent to proving both that finding a Hamiltonian Path from u' to v' gives us a Hamiltonian Cycle and that any Hamiltonian Path in G' must start u' and v'. This is trivial as by construction it can be seen that the Hamiltonian Path P in G' can be transformed into a Hamiltonian Cycle C in G by deleting the starting and end vertices and their adjacent edges i.e. delete u' and v' while deleting edges (u', u) and (v, v'). We have now obtained a Hamiltonian Path in the original Graph G. This can be converted to a Hamiltonian Cycle C by connecting the start and end with edge e = (u, v). But also since u' and v' are vertices with degree 1 (by construction), it follows that any Hamiltonian Path P in G' must start at u' and end at v' (or vice-a-versa), but this implies that checking for a Hamiltonian Path P in G' in general is equivalent to checking for a Hamiltonian Path P in G' from u' to v'
Proving that there does not exist a Hamiltonian Path P in any G' $\implies$ *there does not exist a Hamiltonian Cycle C in G.* However, since the Hamiltonian Cycle C must include some edge e $\in$ E, this is equivalent to proving that if for every edge e = (u, v) there is no Hamiltonian Path P in G' that starts at u' and ends at v' $\implies$ there is no Hamiltonian Cycle in G that contains edge e, and hence there is no Hamiltonian Cycle C in G. But the statement there is no Hamiltonian Path P in G' that starts at u and ends at v is stronger than there being no Hamiltonian Path P in G', hence our blackbox for Hamiltonian Path helps us determine that there is no Hamiltonian Path P in G' that starts at u' and ends at v'. However, the absence of a Hamiltonian Path P in G' from u' to v' implies the absence of a Hamiltonian Path starting at u and ending at v in G but that in turn implies that there can be no Hamiltonian Cycle in G containing e. QED

(2) Reducing Hamiltonian Path to Hamiltonian Cycle

Algorithm:
The general intuition for this reduction lies in the fact that Hamiltonian Paths can always trivial be obtained from a Hamiltonian Cycle, but the other needn't necessarily be true. To reiterate, the notion of having a Hamiltonian Cycle is stronger than the notion of having a Hamiltonian Path. This necessitates that while if we find a Hamiltonian Cycle in G directly we are sure to find a path, but not finding one doesn't necessarily imply no Hamiltonian Path exists. Hence we must augment the graph in a certain way to 'connect' the potential cycles and check all possible combinations of these to convince ourselves that no Hamiltonian Path exists by testing only for Hamiltonian Cycles. Thus formalizing this idea:

for edge unordered pair of vertices (u, v) $\notin$ E i.e. for every pair of vertices u and v that don't have an edge between them **do**
    Let G' $\Leftarrow$ G with an edge e = (u, v) added to it
    if there exists an Hamiltonian Cycle in G' (using blackbox for Hamiltonian Cycle) **then**
        Return True
Return False

Running time:
The for loop runs once for every unordered pair of vertices hence runs for $O(|V|^2)$ time which is polynomial, thus the reduction is poly-time.

Correctness:
Proving that there exists a Hamiltonian Cycle C in a G' $\implies$ there exists a Hamiltonian Path in G There are two cases in which this occurs. Either the new edge e = (u, v) is part of the Hamiltonian Cycle in G' or it isn't. If it isn't disconnecting removing any one edge from the Hamiltonian Cycle in G' arbitrarily gives a Hamiltonian Path in G. If e is part of the Hamiltonian Cycle in G', then removing edge e which doesn't actually exist in G, gives us a valid Hamiltonian Path in G. QED.
Proving that there does not exist a Hamiltonian Cycle C in any G' $\implies$ there does not exist a Hamiltonian Path P in G. If a Hamiltonian Path P were to exist in G, it must start at some vertex u and end at some vertex v. Hence the statement that that no Hamiltonian Path P exists in G is equivalent to saying that no Hamiltonian Path in G starting at u and ending v exists for any u, v, $\in$ V. Now proving that checking for Hamiltonian Cycles by adding new edges (u, v) for every pair of vertices to the graph is equivalent to checking for a Hamiltonian Path starting at u and ending at v. This is true evidently as the Hamiltonian Cycle in a certain G' can be thought of as a concatenation of a Hamiltonian Path in G starting at u and ending at v with the new edge (u, v). Hence no Hamiltonian Cycle in a G' = G + (u, v) implies no Hamiltonian Path P in G starting at u and ending at v, therefore no Hamiltonian Cycle C in any G' implies no Hamiltonian Path P in G. QED

4 Problem 4 **25 / 25**

- ✓ **- 0 pts** Correct
- **- 25 pts** No answer found