Homework 11

Q1: Given an undirected graph G = (V, E), and positive integer k, the max-degree-spanning-tree problem asks whether G has a spanning tree whose degree is at most k. The degree of a spanning tree T is defined as the maximum number of neighbors a node has within the tree (i.e., a node may have many edges incident on it in G, but only some of them get included in T). Show that the max-degree-spanning-tree (MDST) problem is NP-complete. (20pts)

Solution: First we need to prove MDST is in NP. A subgraph can be given as a certificate. The certifier can certify in polynomial time whether this is 1) a tree, 2) spanning all the graph, and 3) the degrees of its vertices within the tree are all less equal k. Hence the problem MDST $\in$ NP.

Then we need to prove MDST is NP-Hard. We use Hamiltonian Path (HP) problem to prove this, which is a known NP-complete problem. We note that a Ham-Path will visit each vertex of the graph exactly once, which will have a max degree of 2 for each vertex in the path. To prove that HP $\leq_p$ MDST, we configure the MDST problem as follows: Given the input of HP - a graph G, we input the exact graph into MDST problem and set K=2.

Now, G has a ham-path if and only if G has a spanning tree of max degree 2. Proof:

- Suppose there is a solution to the MDST problem, that means there is a spanning tree that goes through all vertices with degree less or equal to K=2. It's easy to see there are no 'branches', as no point has 3 connections, hence this tree is simply a path that goes through all vertices, i.e., a Ham-Path in G.
- Vice-Versa, if there is a Ham-Path in G, the path is a tree and it's spanning since it contains all the vertices, and has max degree less than or equal to 2, making it a solution to the MDST problem.

Rubric:

5 points for showing NP: Certificate (2) and certifier (3)

15 points for showing hardness.

- Construction with explanation (8 points)
- Claim for correctness (1 point)
- 3 points for each direction of proof

Q2: The k-cycle-decomposition problem for any $k > 1$ works as follows. The input consists of a connected graph $G=(V, E)$ and k positive integers $a_1, \ldots, a_k < |V|$. The goal is to determine whether there exist k disjoint cycles of sizes $a_1, \ldots, a_k$ respectively, s.t., each node in V is contained in exactly one cycle. Show that this problem is NP-complete (for any $k > 1$). (20 pts)

Solution:
Clearly this problem is in NP. The certificate will be a sequence of k cycles in the graph, and the certifier will check whether each cycle consists of valid edges in the graph, and if cycle i is of length a_i and that each vertex appears in exactly one cycle.

We will reduce HAM-CYCLE to this problem. Given an instance of HAM-CYCLE with input graph $G=(V,E)$, construct a new graph $G'=(V', E')$ by first making k copies of G: G_1 through G_k. Next, connect one arbitrary vertex of G_i to one in G_{i+1} for each $i < k$, so as to make G' a connected graph. set each $a_i = n$ (i.e. $|V|$).
Now, G has a hamiltonian cycle iff G' has a k-cycle-decomposition with each cycle of size n.
Proof:
=>) If G has a Hamilton cycle C, then consider the corresponding copy C_i in each G_i. These cycles are disjoint, have size n each and contain all the vertices in G' by construction. Hence G' has a k-cycle-decomposition as required.
<=) Suppose G' has a k-cycle-decomposition as required. Note that any cycle in G' cannot go across multiple G_i's because the edges that connect the copies do not facilitate it. Hence each of the cycles must be within a single corresponding copy. Since each of these is of size n, it is a Hamilton cycle of the corresponding G_i. As that is merely a copy of G, G must have a Hamilton cycle.

Rubric:
5 points for showing NP. 2 for certificate, 3 for certifier.

15 points for showing NP-hardness

- 9 points for construction.
- 1 point for correctness claim
- 2 points for forward proof (if HC then k-cycle-decomposition)
- 3 points for backward proof (if k-cycle-decomposition then HC)

Q3. Given a graph $G = (V, E)$ with an even number of vertices as the input, the HALF-IS problem is to decide if G has an independent set of size $|V| / 2$. Prove that HALF-IS is in NP-Complete. (20pts)

Solution:

(a) Given a subset of vertices $S \subseteq V$ as certificate, the certifier can check that $|S| = |V|/2$, and that each pair in S is not adjacent (i.e., connected by an edge). This can be done in time ($O(|S|^2) = O(|V|^2)$) i.e. polynomial time. Therefore, HALF−IS $\in$ NP.

(b) We prove HALF-IS is in NP-Hard by using a reduction of the NP-complete problem Independent set problem (IS) to HALF−IS, i.e., IS $\leq$p HALF-IS. Consider an instance of IS, which asks for an independent set $A \subseteq V$, $|A| = k$, for input graph G(V, E). We construct graph $G' = (V', E')$ as instance for HALF-IS differently for 3 different cases. In each case, we show that G has an Ind-Set of size k if and only if G' has an ind-set of size $|V'|/2$. Consider the cases below:

i. If $k = |V|/2$, this IS instance is the same as HALF-IS on $G' = G$.

ii. If $k < |V|/2$, then add m new <u>disconnected (isolated)</u> nodes to get the modified set of nodes V' ($= V \cup \{m \text{ new nodes}\}$). We choose m such that $k + m = |V'|/2 = (|V| + m)/2$, i.e., $m = |V| - 2k$. That makes $|V'| = 2|V| - 2k$, thus an even number. (Note that m is > 0 for this case as $k < |V|/2$).

Proof of the reduction claim: i) If G has an independent set S of size k, we can add all the newly added m nodes to S to get S′. Since the latter are all disconnected from each other and from all nodes of G, S′ is an ind-set in G′. Size of S′ is $k+m = |V'|/2$ by construction. ii) Suppose G′ has an independent set S′ of size $|V'|/2 = k + m$. Remove any of the newly added nodes from S′ to get S. S must have at least k nodes, and $S \subseteq V$, and is an independent set since S′ was. Thus G does have an ind.set of size at least k.

iii. If $k > |V|/2$, then again add m new nodes to form the modified set of nodes V′. This time, Connect these new nodes to each other as well as to all nodes in V. Choose m so that $k = |V'|/2 = (|V| + m)/2$, i.e., $m = 2k - |V|$. That makes $|V'| = 2k$, thus an even number. (Note that m is > 0 for this case as $k > |V|/2$).

Proof of the reduction claim: i) Suppose G has an independent set S of size k, S is also an independent set of G′ and has size $k = |V'|/2$. ii) Suppose G′ has an independent set S′ of size $|V'|/2 = k$. If S′ has any one of the newly added m nodes, say v, it can have no other nodes as they are all connected to v. Therefore, S′ has all the nodes that are in V, making it an ind.set of size k in G. (This argument leaves the corner case $k = 1$, when this one node in S′ can indeed be one of the new nodes, but then G has an ind.set of size 1 anyway as long as it's not empty, so the claim holds. It fails when G is empty so this should be checked outside of invoking the blackbox, to make the proof completely rigorous. No penalty if this subtlety is overlooked.)

Hence, any instance of IS (G(V, E), k), can be reduced to an instance of HALF-IS (G'(V ', E')). This completes the reduction, and we confirm that the given problem is NP-Hard.

Thus this problem is NP-Complete.

Rubric: 4 points for Showing NP. 6 points for describing the constructions spanning the 3 cases. 1 point for correctness claim(s), 1+4+4=9 points for correctness proofs of each of the 3 cases.

Ungraded Problems

Q4. In a certain town, there are many clubs, and every adult belongs to at least one club. The town's people would like to simplify their social life by disbanding as many clubs as possible, but they want to make sure that afterwards everyone will still belong to at least one club.

Formally the Redundant Clubs problem has the following input and output.

INPUT: List of people; list of clubs; list of members of each club; number K.

OUTPUT: Yes if there exists a set of K clubs such that, after disbanding all clubs in this set, each person still belongs to at least one club. No otherwise.

Prove that the Redundant Clubs problem is NP-Complete. (20pts)

Solution:

(a) We must show that Redundant Clubs is in NP, but this is easy: if we are given a set of K clubs, it is straightforward to check in polynomial time whether each person is a member of another club outside this set.

(b) We prove Redundant Clubs is in NP-Hard by reducing from a known NP-complete problem, Set Cover, e.g., Set Cover ≤p Redundant Clubs. We translate inputs of Set Cover to inputs of Redundant Clubs, so we need to specify how each Redundant Clubs input element is formed from the Set Cover instance. We use the Set Cover's elements as our translated list of people, and make a list of clubs, one for each member of the Set Cover family. The members of each club are just the elements of the corresponding family. To finish specifying the Redundant Clubs input, we need to say what K is: we let K = F − KSC where F is the number of families in the Set Cover instance and KSC is the value 'K' from the set cover instance. This translation can clearly be done in polynomial time (it just involves copying some lists and a single subtraction). Finally, we need to show that the translation preserves truth values. If we have a yes-instance of Set Cover, that is, an instance with a cover consisting of KSC subsets, the other K subsets form a solution to the translated Redundant Clubs problem, because each person belongs to a club in the cover.

Conversely, if we have K redundant clubs, the remaining KSC clubs form a cover. So the answer to the Set Cover instance is yes if and only if the answer to the translated Redundant Clubs instance is yes. This completes the reduction, and we confirm that the given problem is NP-Hard. Thus this problem is NP-Complete.

Rubric (20pt):

- 5 pts for Proving Redundant Clubs is in NP.

- 15 pts for Proving Redundant Clubs is in NP-Hard.

– 8 pts for the construction.

– 1 pt for the claim.

– 6 pts for the reduction proof.

Q5. You are given a directed graph G=(V,E) with weights on its edges $e \in E$. The weights can be negative or positive. The Zero-Weight-Cycle Problem is to decide if there is a simple cycle in G so that the sum of the edge weights on this cycle is exactly 0. Prove that this problem is NP-complete. (20pts)

Solution: Zero-weight-cycle is in NP because we can exhibit a cycle in G, and it can be checked that the sum of the edge weights on this cycle are equal to 0. We now show that subset sum $\leq$ Zero-weight-cycle. We are given the number w1, ..., wn, and we want to know if there is a subset that adds up to exactly W. We construct an instance of the Zero-weight-cycle in which the graph has nodes 0, 1, 2, ..., n,n+1 and an edge (i, j) for all pairs $i < j$. The weight of the edge (i, j) is equal to wj for $j < n+1$, and is 0 for $j = n+1$. Finally, there is an edge (n+1, 0) of weight $-W$.

We claim that there is a subset that adds up to exactly W if and only if G has a zero-weight-cycle. If there is such a subset S, then we define a cycle that starts at 0, goes through the nodes whose indices correspond to elements in S (in increasing order of indices), then jumps to n+1 from the last such node and then returns to 0 on the edge (n+1, 0). The weight of $-W$ on the edge (n+1, 0) precisely cancels the sum of the other edge weights since the elements in S summed to W. Conversely, all cycles in G must use the edge (n+1, 0), and so if there is a zero-weight-cycle, then the other edges must exactly cancel $-W$, in other words, their indices must form a set that adds up to exactly W.

Rubric (20pt):

- 5 pts for Proving Zero-weight-cycle is in NP.

- 15 pts for Proving Zero-weight-cycle is in NP-Hard.