Practice Set 14: solutions

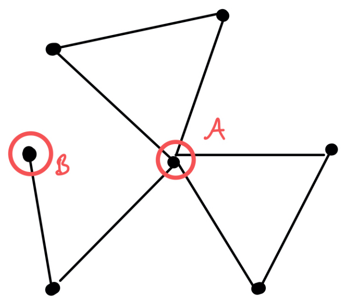
.

Problem 1

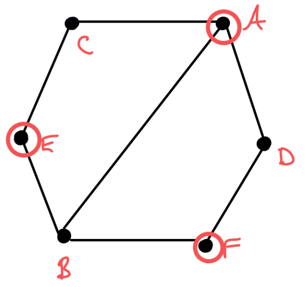
RSA encryption is the algorithm used by modern computers to encrypt and decrypt messages. The receiver creates a public key (n, and e) and "shares" them over a possible non-secure network. The receiver has a private key (d) that is not publicly shared. The sender uses the public keys to encrypt the message, and the receiver uses the private key to decrypt the message. No one worries that a listener who intercepts n and e will be able to determine the private key d. If they could, they would be able to decrypt the message! The reason is that the public key is the product of two very large prime numbers, $n = pq$, and the private key d can only be determined if you knew what those prime numbers were. However, the problem of taking a number n and factoring it into its primes has *no known polynomial time algorithm.* The complexity of this problem is not known. (Note: *there are polynomial-time algorithms that already exist for this problem using quantum computers, in particular Shor's algorithm, meaning a quantum computer would be able to decrypt all RSA messages*).

Problem 2

This algorithm is what is called a "greedy" algorithm. It selects vertices one at a time, by selecting the vertex that has the most neighbours. This intuition may seem to produce the smallest dominating set, because the vertex with the most neighbours "covers" as many vertices as possible. In the example below left, indeed, the process choses two vertices, and this is the smallest dominating set. However, in the example below right, the algorithm selects vertex A and then vertices C, D, B are already covered. Next it selects E, and then F, for a dominating set of size 3. However, a smaller dominating set exists: vertices D and B.

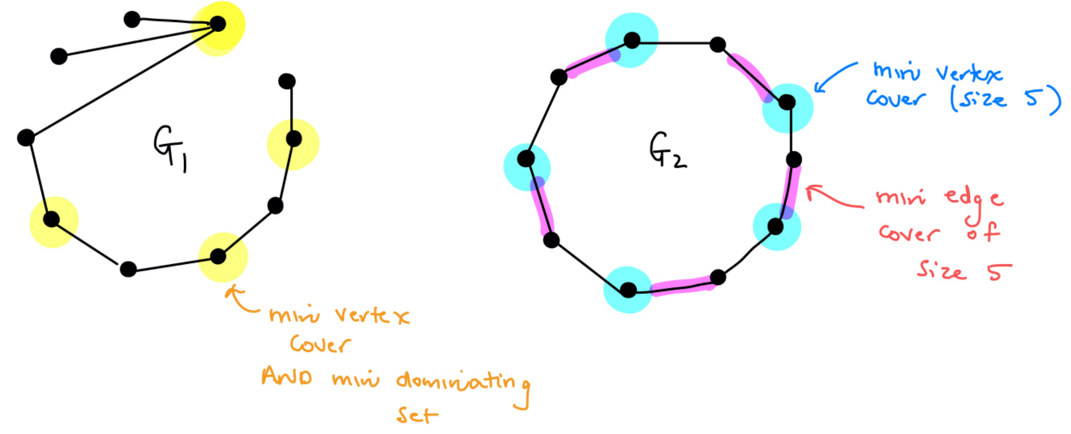


Algorithm picks these 2 vertices as the dominating set. Indeed, this is the smallest dominating set.



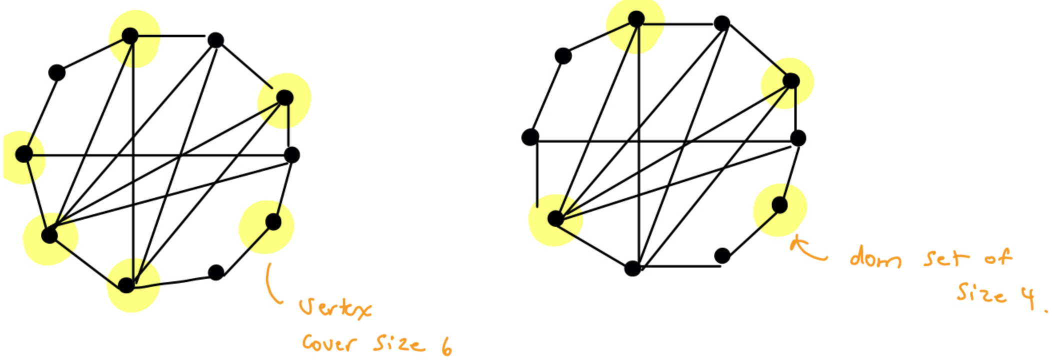
Algorithm picks these 3 vertices as the dominating set. But a smaller set exists.

Problem 3:



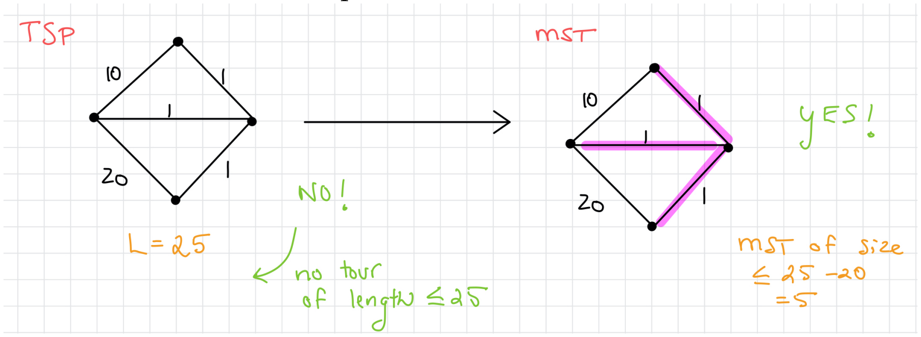
G_1
mini vertex cover AND mini dominating set
G_2
mini vertex cover (size 5)
mini edge cover of size 5

GRAPH 3 :



Problem 4

The reduction takes input to the TSP problem and converts it to input to the MST problem. The amount of time this takes is polynomial in the input size. This follows the definition of a reduction. Next, we need to check if a "yes" instance to TSP would be equivalent to a "yes" instance to MST. Unfortunately, if there is a MST of length at most $L - m$, this does **not** necessarily imply that there is a tour of length at most L. A counter example is shown below:



Problem 5

Step 1: Show that FINDTEAMS is in NP: Suppose we are given a set of n people and the friend relationships, and given two teams T_1 and T_2. Our job is to verify the conditions of FINDTEAMS. First we verify T_1: for every pair of people in T_1, check that that pair represents a friendship. There are $O(n^2)$ pairs of people in T_1. Therefore the runtime is $O(n^2)$. Repeat for T_2. Next check that the number of people in T_1 is the same as the number in T_2, and that they are each $\geq k$. Finally, we check that there are no people in common between T_1 and T_2. The total runtime is $O(n^2)$.

Step 2: Reduction from Clique: The input to Clique is a graph $G = (V, E)$ and in integer k. The goal is to return yes if and only if there is a clique of size k. We show how the Clique problem can be solved with FINDTEAMS. To create the input to FINDTEAMS, we need to define the people and the friendships. We create one person for each vertex in V. For every edge $e = (u, v)$ in E, we create a friendship from person u to v in FINDTEAMS. Next, we create an additional set S of k people. These people will not be friends with anyone in V, but they will all be friends with each other.

Now we verify that the reduction is valid:

1) If G has a clique of size k, then Findteams has two teams of size k

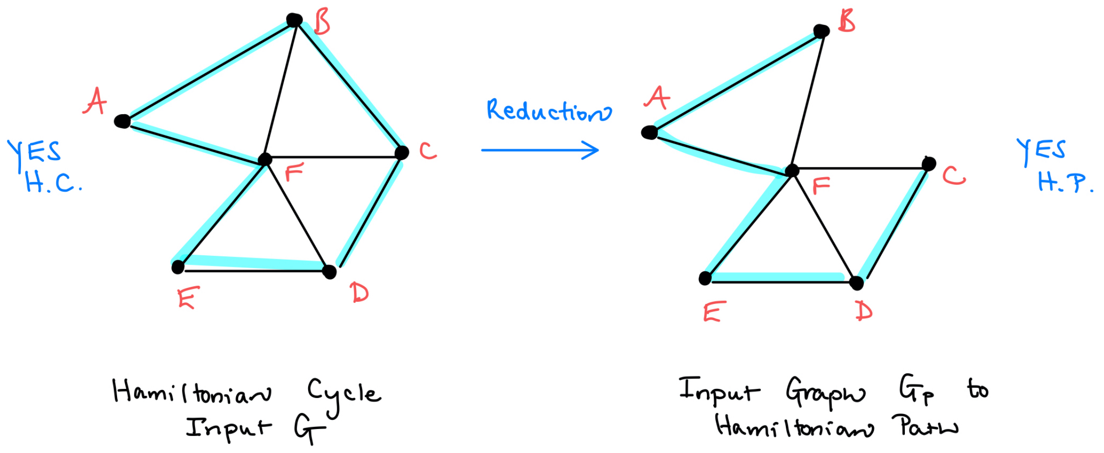
Assume G has a clique of size k. The vertices in this clique correspond to a set of k people forming a team T_1. The second team will be the set of people in set S. Therefore FINDTEAMS returns true.

2) If G does NOT have a clique of size k, then the input to FindTeams will not find two teams of size k:

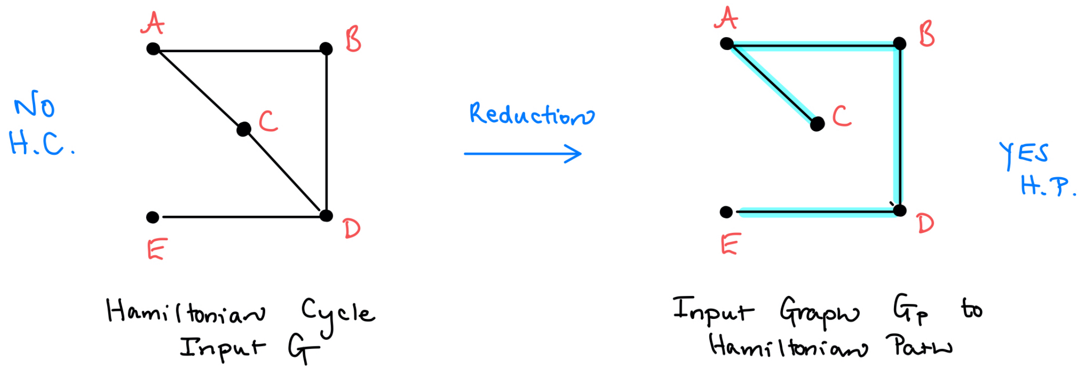
Since G does not have a clique of size k, then the input to FindTeams will NOT have a team of size k in the graph G, but it will have one in the additional set S. Neverthless, FindTeams returns FALSE because there are not TWO teams.

Problem 6:

This may seem like a good idea. In the example below, the input graph G has a Hamiltonian cycle (B,A,F,E,D,C,B), and if we remove an edge (say edge BC) then the new graph G_P has a Hamiltonian Path.



However, the reduction doesn't always work! In the example below, the input graph G does NOT have a Hamiltonian Cycle, but when we remove an edge, the new graph DOES have a Hamiltonian Path. Therefore is reduction is not valid. Nevertheless, Hamiltonian Path is an NP-complete problem. The reduction is different from this one!



Problem 7:

- This problem can be modelled as an undirected graph, where each room is a vertex and a doorway between rooms is an edge that connects two vertices. Selecting cameras in each doorway is equivalent to selecting edges in the graph. Since a doorway camera can observe both adjacent rooms, the selected edge in the graph "covers" both endpoints. So the problem is equivalent to *Edge cover* which we saw in class has a polynomial-time algorithm.

- Again we model the problem as an undirected graph as above. Selecting a camera in a room is like selecting a vertex in the graph. The camera covers the room and the adjacent rooms. Since we want to cover all rooms, this is equivalent to requiring that every vertex in the graph is either selected or adjacent to a selected vertex. Therefore this problem is equivalent to Dominating Set, which we saw in class is NP complete. There is no known polynomial time algorithm for Dominating Set.

Problem 8:

- Again we model the problem as an undirected graph as above. Selecting a room is like selecting a vertex. Since we don't want adjacent rooms selecting, this is equivalent to requiring that no two adjacent vertices in the graph are selected. This is the definition of Independent Set, which is known to be NP complete. There is no known polynomial time algorithm for Independent Set.
- Model the problem as an undirected graph, G. Each person becomes a vertex. We connect an edge between two people if they are **not married nor have been married**. A set of people in a jury would be represented by a set of vertices that all have edges between them. The problem of determining if there is a jury of size at least k is equivalent to determining if the graph G has a clique of size k. Therefore the problem is NP complete.

Problem 9:

- Each webpage is like a vertex, and each link is like and edge in a graph. Clicking through all the webpages is equivalent to Hamiltonian Path, and is therefore NP-complete.
- Model the problem as an undirected graph G, where each of the n vaccine dates is represented by a vertex. Each nurse becomes an *edge* that connects the two vertices that represent her work dates. Since the goal is to hire the minimum number of nurses so that each day has at least one nurse available, this corresponds to selecting the minimum number of edges so that each vertex is connected to at least one selected edges. Therefore the problem is equivalent to *Edge cover* and has a polynomial-time algorithm.

Problem 10

- If we set $k = n - 1$, then this problem is equivalent to the Hamiltonian Path problem, which is NP complete.
- This problem can be solved in polynomial time. There are exactly $\binom{n}{5}$ possible sets of 5 vertices, and for each of those, we can simply check if those 5 vertices form a path. Since $\binom{n}{5}$ is polynomial, this problem is in P.

Problem 11

- The problem is equivalent to Subset Sum, where the water bottle sizes correspond to the items. Subset Sum is NP complete. There is no polynomial-time algorithm known.
- We can model this problem as a graph, where each child is a vertex, and a friendship is an edge between two people. Handing a present to a child is equivalent to selecting that vertex. This problem is equivalent to dominating set, because we need each vertex to either be selected or be adjacent (friends) with a selected vertex. Dominating set is NP complete, and no polynomial-time algorithm is known.

Problem 12

Step 1: Show that we can verify a solution in polynomial time.

Given a set of S of water bottles, we simply add up their weights and verify that the total weight is exactly T. We then add up the total value of each item and show that the total cost is at most C. Each of these summations can be done in polynomial time.

Step 2: Reduction from Subset Sum:

The input to Subset sum is a set of integers and a target sum T. We let these integers be the weights of the water bottles, and the target sum to T. We set each water-bottle cost to 1 and set $C = n$. By doing this, no matter what water bottles are chosen, their total cost will always be at most C. Therefore the selection of water bottles is only constrained by selecting exactly T litres.

Show the reduction is valid:

If there is a solution to the subset sum, then there is a solution to the water bottle problem

If there is a selection of items with total sum T, then by selecting the corresponding water bottles, we have exactly T litres, and the total cost is at most C.

If there is a solution to the water bottle problem, then there is a solution to the subset sum problem

If there is a selection of water bottles with sum exactly T and total cost $\leq n$, then those water bottles correspond to a subset with weight T.

Problem 13

Step 1: Show that a solution to this problem can be verified in polynomial time: Suppose we are given a set of chosen vaccine sites, say C. First verify that $|C| \leq k$. Next, we need to loop through the set of people and verify that their list of acceptable vaccine sites has at least one site in C. There are n people and each person has a list of length $O(m)$, therefore the verification step takes time $O(mn)$.

Step 2: Show this problem is NP complete using a reduction from Vertex Cover: An instance of vertex cover consists of a graph $G = (V, E)$ and an integer k. We convert this into an instance of the vaccine problem. Each edge e in E becomes a person, e, in the vaccine problem. Each vertex $v \in V$ becomes a vaccine site in the vaccine problem. The value of k stays the same. For each edge $e \in E$ we let the two endpoints of e be the two vaccine sites that person e is willing to visit. In other words, for $e = (u, v)$, person e is willing to visit sites u and v. We have now created an instance to the Vaccine problem, where we have a set of people and sites, and each person has a list of exactly two potential vaccine sites. The decision problem is to determine if we can open at most k sites and be able to service all people.

We now verify that the reduction is valid.

1) If there is a vertex cover of size $\leq k$, then it is possible to select $\leq k$ vaccine sites:

Suppose G has a vertex cover called set C of size $\leq k$. Each edge in G is adjacent to at least one vertex in C. Let the vertices in C corresponds to the selected vaccine sites in the vaccine problem. Since each edge $e = (u, v)$ in G corresponds to a person in the vaccine problem with vaccine availabilities u and v, then we know that either u or v is in C, and therefore person e has access to a selected vaccine site.

2) If it is possible to select $\leq k$ vaccine sites, then a vertex cover of size $\leq k$ exists in G.

Suppose it is possible to select a set S of at most k vaccine sites so that each person has access to at least one of their preferred sites. Let the vaccine sites in S correspond to the vertices in the vertex cover of G. Since each person in the vaccine problem was created from an edge $e = (u, v)$ in G, then we know that either vaccine site u or v is in S. Therefore each edge e in G has an endpoint in S. Thus G has a vertex cover of size $\leq k$.

Problem 14

Step 1: Show that a solution to this problem can be verified in polynomial time: Given a solution to FirstAid, we can verify if this solution is valid as follows. For each child, we loop through all the activities of that child, and for each activity loop through all the children in that activity. For each iteration, we check if one of those members has a first-aid kit. The run time is $O(n^2m)$.

Step 2: Show this problem is NP complete using a reduction from Dominating Set: An instance to Dominating Set consists of a graph G and a value k such that the goal is to select at most k vertices such that every vertex is either selected or adjacent to a selected vertex. We create an instance to FirstAid as follows: each vertex v in G becomes a child. Each edge $e = (u, v)$ becomes a activity, where stuents u and v are registered in activity e. The goal is to select at most k students to receive first-aid kids.

1) If there is a Dominating set of size $\leq k$, then there are a most k first aid kits needed: If the graph G has a dominating set of size at most k, then those vertices correspond to students who are receiving first-aid kids. Every child is either a selected vertex, or adjacent to a vertex that is selected. Therefore there a set of at most k first-aid kids is necessary.

2) If at most k kids are necessary, then there is a dominating set of size $\leq k$.

If only k first-aid kits are necessary, then the students who receive those kits correspond to the vertices in G that are selected in the dominating set. Every vertex is either selected, or is adjacent to a selected vertex. Therefore there is a dominating set of size at most k.