

HW10

Sayed Armin Vakil Ghahani

PSU ID: 914017982

CSE-565 Fall 2018

Collaboration with: Sara Mahdizadeh Shahri, Soheil Khadirsharbiyani,
Muhammad Talha Imran

December 5, 2018

Problem 1. Diverse Customers

Solution We can reduce this problem to the Independent Set Problem in the following way. Suppose that we create a graph with n vertices that n is the number of customers. Now, we connect two vertices in this graph if and only if these two customers both have bought a same product. If we have an independent set of size k in this graph, then it means that we have a diverse subset of size k . This means that Diverse Subset problem is NP.

Now, we are going to proof that Independent Set \leq_P Diverse Subset. Suppose that we have a graph of size n with m edges that we want to know it has a independent set of size k or not. We create an array with n customers and for each edge in this graph we create a product. The value of row i and column j in this table is one if and only if vertex i is incident to the corresponding edge with product j .

Finally, this graph has an independent set with size of k if and only if the created array has a diverse set of size k . If this array has a diverse set of size k , these vertices are creating an independent set in the given graph because there is not any edge between them according to the definition of diverse set. Moreover, if the graph has an independent set of size k , these vertices will make a diverse set in the array because none of the pairs has a common product that they have bought before based on the definition.

Problem 2. Four problems with resource allocations

Solution • a) At first, we should note that the certifier of this problem is polynomial because we can check every request is used by just one process or not in $O(nm)$ in easiest way.

Now, we are going to show Independent Set \leq_P Resource Reservation. We can create an instance of Resource Reservation problem from Independent Set problem, and vice versa. The vertices of graph are processes and there is an edge between two processes if and only if these two processes both request a resource.

Finally, if there is an independent set of size k , the vertices corresponding to these processes are disjoint and these processes do not have any share resource in Resource Reservation problem. In addition, if there are k processes that do not share any resource, there is not any edge between the corresponding vertices of these resources in the graph, and it make an independent set of size k .

- b) For this special case, we can check every process and give it all of the resources it wants, and after that, check there is any process that can become active or not in $O(n)$. We can do this for every process, and consequently, solve this problem in $O(n^2)$.
- c) If we draw the same graph in part a in this special case, every vertex in this graph will have at most one edge that is connected to it. So, if we look at each edge, there is not any other edge connected to this edge between if there would be any two neighbour edges, the degree of the vertex in between is 2. As a result, for each edge we just pick one of the vertices, and for the other vertices that do not have edge connected to them, we can collect all of them because there is not any edge connected to them. This independent set is the answer of this special case. We can create this graph in polynomial time, and after that we just check every vertex has a neighbour or not, and in case there is not any neighbour we pick this vertex, and otherwise we pick this vertex and remove its neighbour from the graph. The time complexity of this part is $O(n)$.
- d) This part is the same as part a because we can create the graph in the same way. As a result, the problem is still NP-complete in this speical case.

Problem 3. All Negated or Unnegated in SAT

Solution Suppose that we have n clauses, A_1, A_2, \dots, A_n in SAT problem. We are going to change all the clauses which are not all negate or unnegate. If one of the clauses such as A_i is not all negate or unnegate like $(a \vee b \vee \neg c)$, we change it to $(a \vee b \vee x) \wedge (x \vee c) \wedge (\neg x \vee \neg c)$, where x is a new literal. This new expression will be TRUE iff x and c are not of each other because of the $(x \vee c) \wedge (\neg x \vee \neg c)$. As a result these two expressions are the same as each other. For the clauses that are like $(\neg a \vee \neg b \vee c)$, which their literals are not all negate, we can do the same operation and add a new literal with two new clauses. If we do this operation on all clauses that their literals are not negate or unnegate, we can achieve a special kind of SAT problem and it means that this new SAT problem is also NP-complete.

Problem 4. k-Coloring

Solution • a) 2-Coloring is the same as checking a graph is bipartite or not because we can set each color to each part of the bipartite graph. Moreover, if we our graph is not bipartite, it means that there is an odd cycle in the graph, and it is not possible to color an odd cycle with two colors. In addition, checking that a graph is bipartite or not is in P.

- b) Checking that a coloring for a graph in 4-coloring is easy. We should just check the neighbours of each vertex that they have different colors or not, and moreover, coloring is with 4 colors or just 3 colors.

To prove that 4-Coloring is NP-complete or not, we reduce the 3-Coloring problem to 4-Coloring. To do this, suppose that there is a graph G that can be colored with 3 colors. We can add a new vertex to this graph that is connected to all of the other vertices and make a new graph G' . This new graph cannot be colored with 3 colors because the new vertex has a neighbour of each color and it should be colored with a new color. Moreover, it is obvious that this new graph can be colored with 4 colors if G is 3-colorable.

In addition, if G' is 4-colorable we will know that the only vertex with a color number $\neq 4$ is the new vertex that we have added. As a result, if we remove it from the graph we will have graph G that is 3-colorable.

Thus, 4-Coloring problem is NP-complete.

- c) I want to use 30% option for this question.