**Advanced Algorithms**

**Exercise for Lecture 15**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name** |  | **Student ID** |  |
| **Problem 1** |  | | |
| **Problem 2** |  | | |
| **Problem 3** |  | | |
| **Total Score** |  | | |
| **Notes** | Deadline: **2023-11-13 24:00**  Submission Format: ‘**Lecture15\_Name\_Student ID.docx**’, and please send to: **[chenlq1997@126.com](mailto:algorithms_23fall@163.com)**.  This assignment is meant to be an evaluation of your **individual** understanding coming into the course and should be completed **without collaboration** or outside help. | | |

**Problem 1.[30 points]**

There exists two couriers, they must deliver a number of parcels to a number of addresses. They will go to the express point first every day to get the parcels. And they will go back to the express point at the end of every day to report the work. This gives the following decision problem.

**Instance:** Set *V* of locations. For each pair of locations *v, w ∈ V* , there is a distance *d*(*v, w*) *∈* N, a starting location *s ∈ V* , and an integer *K*.

**Question:** Are there two cycles, that both start in *s*, such that every location in *V* is on at least one of the two cycles, and both cycles have length at most *K*?

Show that this problem is NP-complete.

**Solution:**

If there is only one courier, then the problem above boils down to the decision variant of the TRAVELING SALESMAN PROBLEM (TSP). This gives us the idea to use a reduction from TSP or Hamiltonian Cycle. To be sure that the distances will become non-negative integers, we use a reduction from Hamiltonian Cycle so that in the above problem we can define the distances ourselves, depending on the presence of an edge. The only thing we have to do is to keep the other driver busy, which is easily taken care of by including an additional vertex with distance k/2 from s and distance k+1 to all other addresses.

Hence, we find the following reduction. Given any instance of Hamiltonian Cycle with n vertices, construct the following special instance of the above problem. Copy the graph and make the distance d(u,w) equal to 2 if there is an edge in the graph, and 4, otherwise. Label one of these vertices as s. Add one vertex 0 such that d(s,0) = d(0, s) = n and d(v, 0) = d(0, v) = 2n + l for all v≠s. The threshold K is equal to 2n.

**Proof:**

It is obvious that we can prove this problem is a NP problem since we can verify whether the cycles match the requirements of the problem in polynomial time if the cycles are given. We just need to check whether the nodes of two cycles cover all nodes in V. And that will cost at most O(N+2k). So it is a NP problem.

TSP is a NPC problem. The reduction function T is: Given a instance of Hamiltonian cycle problem with n vertices. Copy the graph and set the distance d(v,w) as 2 if v is connected to w and 4 if v is not connected w. Label one vertex of these vertices as s and add one vertex with label 0. d(0,s)=n and d(0,v)=2n+1 where v is any other vertex in V. K equals 2n. Since there exist one point with label 0 and its distance to s is n, which is equal to k/2. That means one trunk can only go to label 0 point. And the other trunk go to the left points. After function T, the Hamiltonian cycle problem reduce the above problem.

The T is in polynomial time

O()

So it is a NPC problem.

**Problem 2.[30 points]**

Please prove that clique problem is NPC. Given that 3-SAT problem is a NPC problem.

For a given undirected graph G=(V,E), the clique of it is V’ which is a subset of V. For any point pair u,v∈V’, the edge (u,v)∈E. And the clique problem is to find the max clique which contains most points of the graph G.

**Solution:**

Proof:

First this is a NP problem. For any subset V’ of the vertex set V, we can check whether each pair of vertices of V’ is the edge in E. For any u,v∈V’, we just need to check whether (u,v)∈E, it will cost at most O() which is in polynomial time. So we can verify whether the cycles match the requirements of the problem in polynomial time. So it is a NP problem.

And then we can prove this is a NPC problem with the help of 3-SAT problem.

The reduction function T is shown as follows.

Let ɑ=C1 ⋀C2 ⋀C3 ⋀C4 ⋀…… ⋀Ck. For any Ci, it contains 3 words ,. We will construct the graph G as follows:

For any Ci=(), we will set three vertices in the graph. If the pair of vertices, like , meet the following conditions, we will link them in the graph G.

1. r≠s

Then we get the undirected graph G. To make ɑ have satisfiability assignment, there is sure to exist a 1 assignment for three words in each Ci. For each Ci, we choose only one word to assign value 1 and the corresponding vertex is included in the vertex subset V’. And the V’ is the max clique for the graph G. That is because if any other vertex of Ci , for example, is included in V’, there will exist one pair of vertices in V’ which are not linked since in Ci, the vertices are not linked to each other and there has existed one vertex of Ci in V’.

So the 3-SAT problem is reduced to the clique problem. And the function T is

O() = O(n2)

Which is in polynomial time.

So the clique problem is NPC.

**Problem 3.[40 points]**

Please prove that vertex cover problem is NPC. Given that clique problem is a NPC problem.

For a given undirected graph G=(V,E), the vertex cover of it is a V’ which is a subset of V. For any edge (u,v)∈E, the point u∈V’ or v∈V’(or both u,v∈V’). That is to say for every edge that belong to E, at least one point of the edge belongs to the vertex set V’, and then V’ is a vertex cover of G. The vertex cover problem is to find a min vertex cover which contains least points of the graph G.

We are given an undirected graph (*V, E*). A vertex cover is a subset *W* ⊆ *V* such that for each (*v, w*) *∈ E* we have *v ∈ W* or *w ∈ W*. We consider the following problem. Vertex Cover

**Instance:** Undirected graph *G* = (*V, E*), integer *K*.

**Question:** Does *G* have a vertex cover of at most *K* vertices?

3.1 Show that Vertex cover belongs to the class NP.

3.2 Proof that the Vertex Cover problem is NP-complete by a reduction from max clique problem.

**Solution:**

3.1 A solution is a subset V' of the nodes and can be encoded in O(n). Checking if a solution is a feasible solution, you have to check for each edge if it has an end point in V which requires O(n2) time. So the time is in polynomial time. So it is a NP problem.

3.2 The reduction function T is:

For an undirected graph G, we define the supplementary graph Gsup = (V, ~~E~~), where Esup={(u,v)|u,v∈V, u≠v, (u,v)∈E}. It is obviously that for a clique V’ of G, the V - V’ is sure to be a vertex cover of Gsup. We let (u, v)∈~~E~~, and that means (u,v)∉E. So at least one of u and v is not included in V’. So at least one of u and v is included in V - V’. And that means (u,v) is covered by V-V’ in Gsup. So for any edge (x,y) in graph Gsup is sure to be covered by V-V’ . So the clique problem is reduced to vertex cover problem. And it is the same that, if V-V’ is a vertex cover of Gsup, then V’ is sure to be the clique of G. So, to get the min vertex cover of G, we just need get the max clique of Gsup. So the max clique problem is reduced to min vertex cover problem.

The reduction function is in polynomial time of

O() = O(n2)

So the vertex cover problem is NPC.