NYO-China Interview Report

Name: Wayne E-mail: beyongfengwei@gmail.com

Part1 Find Duplication in array.

This is a really simple but interesting question. There are many different ways to solve this problem.

I implement different ways we discussed in the interview to solve the problem.

More details are in the comment of source code.

Part2 K-Means on undirected Graph

K-means is a classic clustering algorithm in machine learning, usually it is applied with the Point such as (x, y), It is my first time use it with the graph.

K-Mean algorithm:

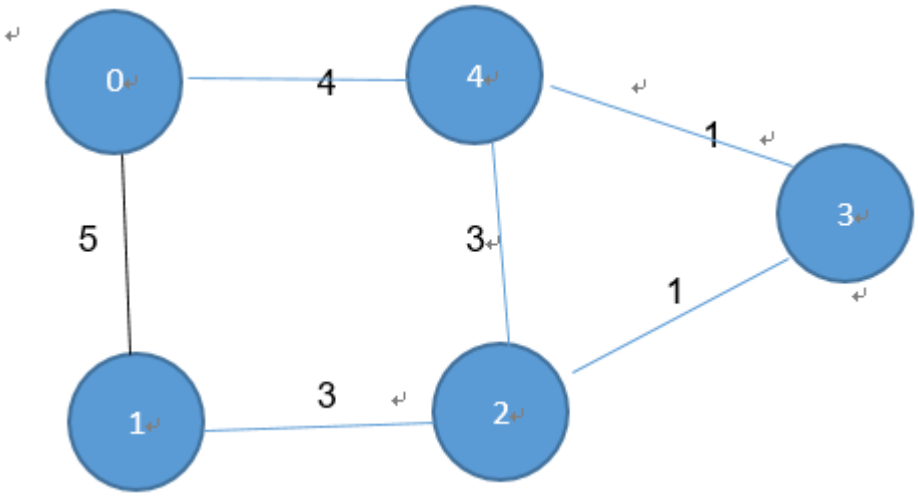
1) Select N random point as center points from the dataset.
2) Calculate the distance between each point and center points, and assign it to one cluster
3) Update the center points.
4) Get the distance between last center points and current center points, if the distance is smaller than threshold, the algorithm is over.

For step 1: It is easy to select n random vertex from dataset.

For step 2: In the graph, I use dijkstra to get the shortest distance from source vertex and center points, then assign it to one cluster.

For step 3: update the center points is not easy. Then select each node of cluster as root to create a minimum spanning tree. Then calculate the path sum of the tree, update the final center with smallest path sum.

For step 4: If the center point keep same, the algorithm is over.

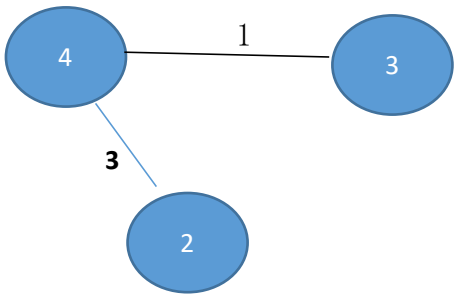


We can use the graph above as the example.
Step 1: If the select node 0 and node 4 as the initial cluster center of cluster 0 and cluster 1.

Step 2: For node 1, the shortest distance from node 0 is 5， the shortest distance from node 4 is 3 + 3 = 6. So node 1 belongs to cluster 0. Similarly, node 4 and node 2 belongs to cluster 1.

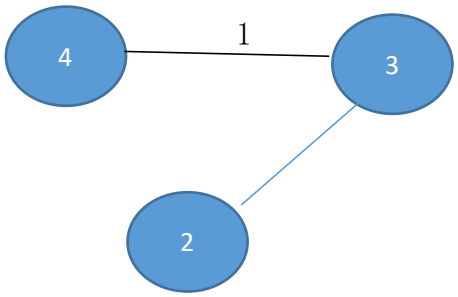
Step 3: now cluster 0 has two nodes, node 0 and node 1, then select each node as root to create a minimum spanning tree. The sum of path is same. So the center of cluster 0 is node 0.

Cluster 1 has three nodes, node 2, node 3, node 4. The current cluster center is 4, when select four as root we can get the minimum spanning tree is like this.



The sum of path is 4, which is as same as selecting node 2 as the root.

If we select 3 as root, the tree is like:



The sum of path is 2. So we update the center of cluster 1 from node 4 to node 3.

Step 4: Finally, we can find node 0 and node 3 are the center points of two cluster.

There is another algorithm from stack overflow

```
1. for i = 1..k do // for each cluster
2.     choose the number of nodes N in cluster i
3.     choose an arbitrary node n
4.     run breadth-first search (BFS) from n until N
5.     assign the first N nodes (incl. n) tapped by the BFS to the i-th
cluster
6.     remove these nodes (and the incident edges) from the graph
7. done
```

http://stackoverflow.com/questions/26521784/how-to-partition-the-nodes-of-an-undirected-graph-into-k-sets

However, I think it calculate how many nodes of one cluster and then assign n shortest one with BFS. It is too simple to get a good partition. So, the use the path sum of minimum spanning tree as the cost function.

When I build the minimum spanning tree, I use union Find class instead of priority queue. Union find is more effective than priority queue.