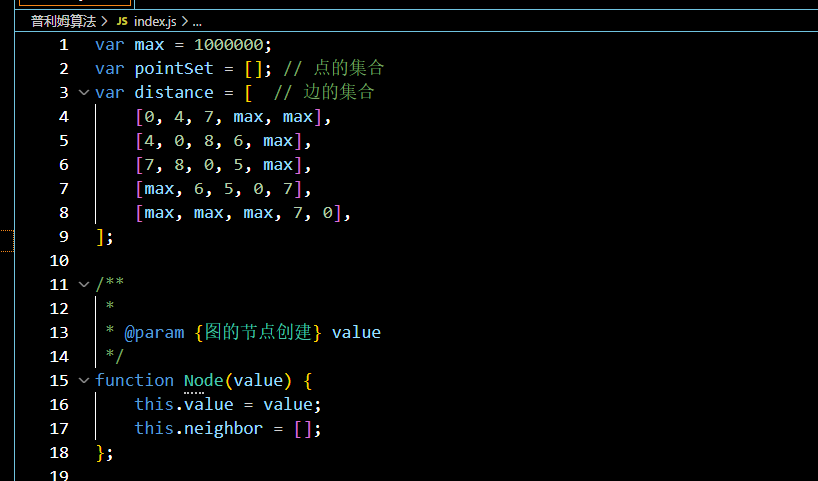
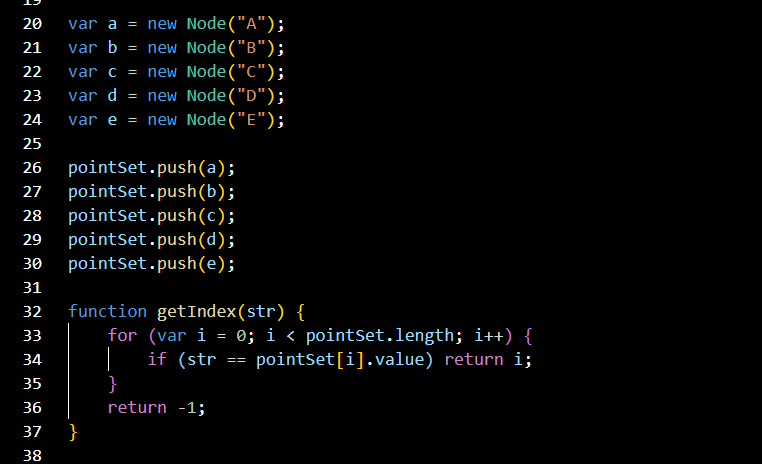
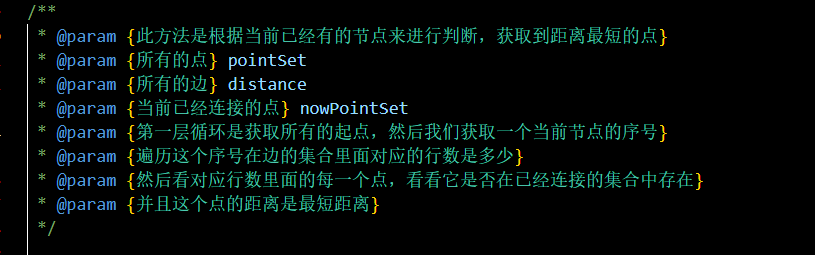
普利姆算法(加点法)

代码实现：

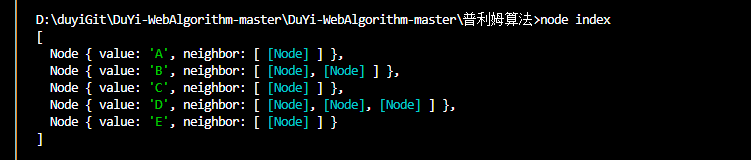














我明白你的困惑。在这个方法中，通过使用双层for循环来遍历distance数组，每次比较distance[noPointIndex][j]和minDic。在每次比较中，minDic都会保留当前已知的最小距离。这个方法的目的是找到当前节点集合与未包含的节点之间的最小距离，以便在最小距离的节点与当前节点之间建立连接。

由于minDic初始值为max（非常大的值），因此第一次比较时minDic必然会被更新为distance[noPointIndex][j]，接下来的比较会将minDic更新为更小的距离。在方法的最后，minDic会变成当前节点集合与未包含节点之间的最小距离

特性和理论逻辑：

1. 任选一个点作为起点
2. 找到以当前选中的点为起点路径最短的边
3. 如果这个边的另一端没有被连通进来，那么就连接彼此，如果这个边的另一端也早就被连进来了，则看倒数第二短的边
4. 重复2-3直到将所有的点都连接为止。
5. 以某一个村庄C为起点看他修哪条路用的钱最少，就修哪条路，比如修DC这条路，修完之后从DC开始看修哪条路的钱最少，得出DB才花6W修B然后再看，CDB开始修哪条路用的钱最少，AB，那么就修AB，然后再看ABCD修哪条路用的钱最少BC是8W ，E才7W而且它是没连通的所以连E为什么不连AC呢？因为你绕一下能到达AC就没必要浪费钱了。

B3

.

6

7

5

8

7

4

E5

D2

C1

A4