

## 华东师范大学数据科学与工程学院上机实践报告

课程名称：算法设计与分析

年级：19 级

上机实践成绩：

指导教师：金澈清

姓名：龚敬洋

上机实践名称：求最大团

学号：

上机实践日期：

10195501436

2021/1/8

上机实践编号：No. 13

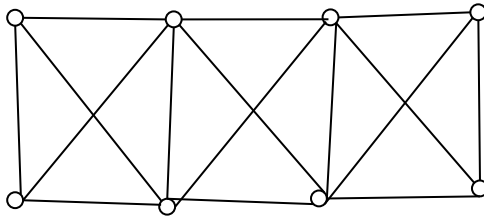
组号：1-436

### 一、目的

1. 熟悉算法设计的基本思想
2. 理解 NPC 问题的难度
3. 求图中最大团

### 二、内容与设计思想

给定一个图，如下所示。请找出这个图的最大团。



### 三、使用环境

推荐使用 C/C++ 集成编译环境。

### 四、实验过程

#### 1. 编写相关实验代码

```
1. #include <iostream>
2. #include <fstream>
3. #include <algorithm>
4. #include <set>
5. #include <vector>
6. using namespace std;
7. int adj[10][10] = {0}, n;
8. vector<set<int>> cliques;
9. void find_clique(set<int> r, set<int> p, set<int> x){
10.     if(p.empty() && x.empty()){
11.         cliques.push_back(r);
12.     }
13.     else if (p.empty()){
14.         return;
```

```

15.     }
16.     set<int> p_mut = p;
17.     for(int iter : p){
18.         set<int> r_tmp = r, p_tmp, x_tmp, adj_tmp;
19.         r_tmp.insert(iter);
20.         for (int i = 0; i < n; i++)
21.             if(adj[iter][i] == 1)
22.                 adj_tmp.insert(i);
23.         set_intersection(begin(p_mut), end(p_mut), begin(adj_tmp), end(adj_tmp), inserter(
p_tmp, p_tmp.begin()));
24.         set_intersection(begin(x), end(x), begin(adj_tmp), end(adj_tmp), inserter(x_tmp, x
_tmp.begin()));
25.         find_clique(r_tmp, p_tmp, x_tmp);
26.         p_mut.erase(iter);
27.         x.insert(iter);
28.     }
29. }
30. int main() {
31.     ifstream fin("data.txt");
32.     set<int> r, p, x;
33.     fin>>n;
34.     while(!fin.eof()){
35.         int fe, te;
36.         fin>>fe>>te;
37.         adj[fe][te] = 1;
38.         adj[te][fe] = 1;
39.     }
40.     for(int i = 0; i < n; i++) p.insert(i);
41.     find_clique(r, p, x);
42.     unsigned int max_cs = 0;
43.     for(int i = 0; i < cliques.size(); i++) {
44.         if(cliques[i].size() > max_cs) max_cs = cliques[i].size();
45.     }
46.     for(int i = 0; i < cliques.size(); i++) {
47.         if(cliques[i].size() == max_cs){
48.             for(int iter: cliques[i]){
49.                 cout<<iter<<" ";
50.             }
51.             cout<<endl;
52.         }
53.     }
54.     fin.close();
55.     return 0;
56. }

```

## 2. 写出算法的思路。

使用 Bron-Kerbosch 算法，维护三个不相交的顶点集  $P, R, X$ ，其中  $P$  代表待搜索的元素， $R$  代表当前极大环中的元素， $X$  代表被排除的元素。从图中的某个点开始，递归的遍历所有其邻接顶点，当  $P$  为空集时，若  $X$  也为空集，则表明找到了一个极大团，否则回溯到上一节点，并将其从  $P$  中除去加入  $X$  中。最终即可找到所有极大团，进而比较找出整张图中的最大环。

## 五、总结

对上机实践结果进行分析，问题回答，上机的心得体会及改进意见。

求最大团问题是一个 NPC 问题，即无法确定其是否存在多项式时间算法的可归约问题。Bron-Kerbosch 算法的时间复杂度约为  $O(3^n)$ ，随着数据规模增大运行时间会显著增加。