

# 算法设计与分析 E4-A

21377206 阮阳栋

## 题目描述

给定 $n$ 个点， $m$ 条有向边的有向无环图，求拓扑排序。在不唯一时先输出序号大的点，再输出序号小的点。

## 题目分析

由于需要字典序最大的拓扑排序，可以用BFS来进行求解，并且用优先队列来维护搜索。

## 题目求解

首先需要找入度为0的点来构成搜索的第一组，然后对每点进行遍历并且构建新的队列：

```
void solve(){
    for (int i=1;i<=n;i++)if(!d[i]) h.push(i); //入度为0
    while (!h.empty()){ //bfs
        int u = h.top(); h.pop();
        printf("%d ", u);
        for (int y : g[u]){
            d[y]--;
            if (!d[y]) h.push(y); //入度为0，进队
        }
    }
    puts("");
}
```

重边可以用集合维护：

```
set <int> g[M];
```

读入时直接进集合：

```

for (int i=1;i<=m;i++){
    int u, v; scanf("%d %d", &u, &v);
    g[u].insert(v);
}

```

## 时间复杂度

只涉及计算出度入度以及广度搜索的时间复杂度为 $O(n)$ ，但是有集合和优先队列，最终时间复杂度为 $O(n \log n)$ 。

## 核心代码

```

int d[M];
set<int> g[M];
priority_queue<int, vector<int>, cmp> h;

void solve(){
    for (int i=1;i<=n;i++)if(!d[i]) h.push(i);
    while (!h.empty()){
        int u = h.top(); h.pop();
        for (int y : g[u]){
            d[y]--;
            if (!d[y]) h.push(y);
        }
    }
    cout << endl;
}

```