

C4 - H 2024-TOPO!

DeNeRATe 杨佳宇轩

11/6/2024

题目描述

- 给定一个 DAG，求出该图字典序最大的拓扑排序

解题思路

- 回顾普通的拓扑排序，即使用 `queue` 进行实现时
- 对于当前图状态下，我们仅关心入度归零的点有哪些，为了求字典序最大的拓扑排序，我们可以想到尝试使用 `priority_queue`
- 对于当前的图状态，其入度归零的点是一定的，那么为了使整体的字典序最大，当前次选择的点一定得是最大的，因此，**单次选择的最优解无后效性**，一定是整体的最优解。

解题思路

- 回顾普通的拓扑排序，即使用 `queue` 进行实现时
- 对于当前图状态下，我们仅关心入度归零的点有哪些，为了求字典序最大的拓扑排序，我们可以想到尝试使用 `priority_queue`
- 对于当前的图状态，其入度归零的点是一定的，那么为了使整体的字典序最大，当前次选择的点一定得是最大的，因此，**单次选择的最优解无后效性**，一定是整体的最优解。

解题思路

- 回顾普通的拓扑排序，即使用 `queue` 进行实现时
- 对于当前图状态下，我们仅关心入度归零的点有哪些，为了求字典序最大的拓扑排序，我们可以想到尝试使用 `priority_queue`
- 对于当前的图状态，其入度归零的点是一定的，那么为了使整体的字典序最大，当前次选择的点一定得是最大的，因此，**单次选择的最优解无后效性**，一定是整体的最优解。

代码实现

时间复杂度: $O(n \log n)$

```
1  vvi edge(n + 1); // edge[u].pb(v);
2  vi RD(n + 1), ans;
3  ...
4  priority_queue<int> q;
5  rep(i, 1, n) if(RD[i] == 0) q.push(i);
6  while(!q.empty()) {
7      int now = q.top();
8      q.pop();
9      ans.pb(now);
10     for(auto ed: edge[now]) {
11         RD[ed]--;
12         if(!RD[ed]) q.push(ed);
13     }
14 }
15 wt(ans);
```

Thank you!