

树的最小支配集

Minimum Dominating Set on Tree

Problem: 选出一个点的集合，使得所有不在这个集合中的点都与至少一个在这个集合中的点相邻。

Idea: 贪心。对树 dfs 一遍，按 dfs 序列逆序检查每个节点，若它属于支配集或者已经被支配，则继续；否则将它的父节点放入支配集。

Complexity: $O(n)$

Code:

```
1  int fa[N];
2  stack<int> sta;
3  void dfs(int x, int f){
4      fa[x] = f;
5      sta.push(x);
6      for(int i = head[x]; i; i = edge[i].nxt){
7          if(edge[i].to == f) continue;
8          dfs(edge[i].to, x);
9      }
10 }
11
12 bool inSet[N], beDom[N];
13 void getSet(){
14     while(!sta.empty()){
15         int cur = sta.top(); sta.pop();
16         if(!beDom[cur]){
17             beDom[cur] = 1;
18             if(fa[cur] == 0) inSet[cur] = 1;
19             else{
20                 inSet[fa[cur]] = beDom[fa[cur]] = 1;
21                 if(fa[fa[cur]])
22                     beDom[fa[fa[cur]]] = 1;
23                 for(int i = head[fa[cur]]; i; i = edge[i].nxt){
24                     beDom[edge[i].to] = 1;
25                 }
26             }
27         }
28     }
```

