

 题目

07:41:57



P2_FolderSize

文件上传

题目编号 1323-1511



Folder Size



任务



给定一个项目的目录树，每个节点要么是文件，要么是文件夹。请实现一个 MIPS 程序，支持多次查询：对给定的节点编号，输出其“子树总大小”(该文件夹内所有文件大小之和，包含所有子文件夹内的文件)。



具体要求

输入

第一行：整数 n ，表示节点总数（节点编号为 $1, 2, \dots, n$ ）。

接下来 n 段节点描述（按编号顺序从小到大给出，即第 i 段节点描述对应第 i 个节点）：

- 若为文件：第一行为 0，第二行一个整数 $size$ ，表示文件大小。
- 若为文件夹：第一行 1，第二行一个整数 k 。接下来 k 行依次给出 k 个整数，为该文件夹的子节点编号。

然后一行输入一个整数 q ，表示查询次数。

接下来 q 行：每行一个整数 id ，表示查询该节点为根的子树总大小（保证 id 合法）

约束：

- $1 \leq n \leq 150$ 。
- 文件大小 $size$ 满足 $0 \leq size \leq 10^6$ 。
- 所有文件大小总和 $\leq 10^9$ 。
- 输入保证形成一个或多个有根树（无环）。
- 总边数 ≤ 800 。
- 查询次数 ≤ 60 。

输出

 题目07:41:57  

示例

 输入

5
1
2
2
3
0
 100
1
1
4
0
20
0
1
3
1
3
5

解释：

- 节点 1：文件夹，有两个子节点（2 和 3）。
- 节点 2：文件（大小为 100）。
- 节点 3：文件夹，有一个子节点（4）。
- 节点 4：文件（大小为 20）。
- 节点 5：文件（未被任何文件夹包含，大小为 1）。

查询三次，分别为 1、3 和 5。

输出样例

120
20
1

说明：

- 子树 1 包含文件 2 和 4，总计 120。
- 子树 3 仅包含文件 4，大小为 20。



07:41:57



题目

提交要求



- 请勿使用 .globl main
- 不考虑延迟槽
- 只需要提交 .asm 文件
- 程序的初始地址设置 (Mars->Settings->Memory Configuration) 为 Compact, Data at Address 0



参考代码



```
#include <stdio.h>

#define MAXN 155
#define MAXE 805

int n;
int nodeType[MAXN];      // 0: 文件, 1: 文件夹
int fileSizeVal[MAXN];   // 文件大小(文件夹为0)
int headAdj[MAXN];       // 邻接表头
int toEdge[MAXE], nextEdge[MAXE];
int edgeCnt;

void add_edge(int u, int v) {
    toEdge[edgeCnt] = v;
    nextEdge[edgeCnt] = headAdj[u];
    headAdj[u] = edgeCnt++;
}

int dfs_sum(int u) {
    if (nodeType[u] == 0) {
        return fileSizeVal[u];
    }

    int size = 0;
    for (int e = headAdj[u]; e != -1; e = nextEdge[e]) {
        int v = toEdge[e];
        size += dfs_sum(v);
    }
    return size;
}

int main() {
    scanf("%d", &n);

    for (int i = 1; i <= n; i++) {
        headAdj[i] = -1;
    }
}
```

07:41:57



题目

```
edgeCnt = 0;

for (int i = 1; i <= n; i++) {
    int t, x;
    scanf("%d %d", &t, &x);
    if (t == 0) {
        nodeType[i] = 0;
        fileSizeVal[i] = x;
    } else {
        nodeType[i] = 1;
        int k = x;
        for (int j = 0; j < k; j++) {
            int v;
            scanf("%d", &v);
            add_edge(i, v);
        }
    }
}

int q;
scanf("%d", &q);
while (q--) {
    int id;
    scanf("%d", &id);
    int ans = dfs_sum(id);
    printf("%d\n", ans);
}

return 0;
}
```

提交 P2_FolderSize

 点击/拖拽选择文件

提交