1. **代码实践**

**0919编程测试第四题：图的DFS+DP：**

#include<algorithm>

#include<cstdio>

int tot, go[200005], first[200005], next[200005], val[200005];

int dis[200005], f[2005][2005];

int n, w[200005], d[200005], best[200005];

void insert(int x, int y, int z) { //建图，go中存储节点，val存储代价，first存储节点i最后一次出现时的次序，next用于递归遍历i的临接点。

tot++;

go[tot] = y;

next[tot] = first[x];

first[x] = tot;

val[tot] = z;

}

void add(int x, int y, int z) { //双向图

insert(x, y, z);

insert(y, x, z);

}

void Dfs(int x) { //递归遍历x的临接点

for (int i = first[x]; i; i = next[i]) {

int pur = go[i];

if (dis[pur] != -1) continue;

dis[pur] = dis[x] + val[i]; //pur节点到根节点的总代价

Dfs(pur);

}

}

void dfs(int x, int fa) {

for (int i = first[x]; i; i = next[i]) {

int pur = go[i];

if (pur == fa) continue;

dfs(pur, x);

}

for (int i = 1; i <= n; i++) dis[i] = -1;

dis[x] = 0;

Dfs(x);

best[x] = 99999999;

for (int i = 1; i <= n; i++) f[x][i] = 99999999;

for (int i = 1; i <= n; i++)

if (dis[i] <= d[x]) { //遍历所有符合x节点要求的建馆位置，包括x节点自身

f[x][i] = w[i];

for (int j = first[x]; j; j = next[j]) {

int pur = go[j];

if (pur == fa) continue;

f[x][i] += std::min(best[pur], f[pur][i] - w[i]);

}

best[x] = std::min(best[x], f[x][i]);

}

}

int main() {

int T;

scanf("%d", &T);

while (T--) {

scanf("%d", &n);

for (int i = 1; i <= n; i++) scanf("%d", &w[i]);

for (int i = 1; i <= n; i++) scanf("%d", &d[i]);

tot = 0;

for (int i = 1; i <= n; i++) first[i] = 0;

for (int i = 1; i < n; i++) {

int x, y, z;

scanf("%d%d%d", &x, &y, &z);

add(x, y, z);

}

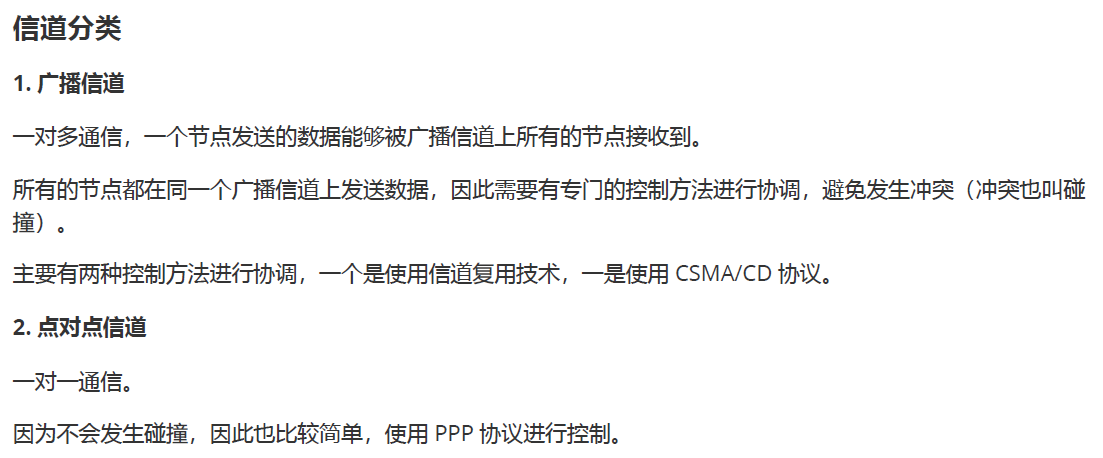
dfs(1, 0);

printf("%d\n", best[1]);

}

}

1. **计算机基础整理**



1. **开源特训营工作总结**
2. 将9月22日每日作业提交到Git仓库中。