

## ● 代码实践

```
#include<stdlib.h>
#include<stdio.h>
#include<string.h>
#include<math.h>
#define N 1005
int w[N];
int d[N];
int map[N][N];
int sum[N];
int n;
int min=100000000;
bool record[N];

void dfs(int start,int mid,int num,int end)
{
    if(map[start][end]>0) return;
    if(mid==end)
    {
        map[start][end]=num;
        map[end][start]=num;
        return;
    }
    for(int i=1;i<=n;i++)
    {
        if(mid!=i&&map[mid][i]>0&&record[mid]==0)
        {
            record[mid]=1;
            dfs(start,i,num+map[mid][i],end);
            record[mid]=0;
        }
    }
}

bool visited[N];

bool istart()
{
    for(int i=1;i<=n;i++)
        if(sum[i]==0) return true;
    return false;
}
```

```

bool isok()
{
    for(int i=1;i<=n;i++)
        if(sum[i]>d[i]) return false;
    return true;
}

void helper(int t,int now)
{
    if(!istart())
    {
        if(isok())
        {
            if(min>now) min=now;
            return;
        }
    }
    for(int i=t;i<=n;i++)
    {
        if(!visited[i])
        {
            visited[i]=1;
            int tmp[N];
            for(int j=1;j<=n;j++)
                tmp[j]=sum[j];
            for(int j=1;j<=n;j++)
            {
                if(sum[j]!=-1)
                {
                    if(sum[j]==0)
                        sum[j]=map[i][j];
                    else
                        sum[j]=sum[j]<map[i][j]?sum[j]:map[i][j];
                }
            }
            sum[i]=-1;
            helper(i+1,now+w[i]);
            for(int j=1;j<=n;j++)
                sum[j]=tmp[j];
            visited[i]=0;
        }
    }
}

```

```

int main()
{
    int T;
    scanf("%d",&T);
    while(T--)
    {
        scanf("%d",&n);
        min=100000000;
        memset(sum,0,sizeof(sum));
        memset(map,0,sizeof(map));
        memset(record,0,sizeof(record));
        memset(visited,0,sizeof(visited));
        for(int i=1;i<=n;i++)
            scanf("%d",&w[i]);
        for(int i=1;i<=n;i++)
            scanf("%d",&d[i]);
        int a,b,l;
        for(int i=1;i<n;i++)
        {
            scanf("%d%d%d",&a,&b,&l);
            map[a][b]=1;
            map[b][a]=1;
        }
        for(int i=1;i<=n;i++)
            for(int j=i+1;j<=n;j++)
                dfs(i,i,0,j);
        helper(1,0);
        printf("%d\n",min);
    }
    return 0;
}

```

## ● 计算机基础知识整理

### 进程与线程

一个在内存中运行的应用程序。每个进程都有自己独立的一块内存空间，一个进程可以有多个线程，比如在 Windows 系统中，一个运行的 exe 文件就是一个进程。

进程中的一个执行任务（控制单元），负责当前进程中程序的执行。一个进程至少有一个线程，一个进程可以运行多个线程，多个线程可共享数据。

与进程不同的是同类的多个线程共享进程的堆和方法区资源，但每个线程有自己的程序计数器、虚拟机栈和本地方法栈，所以系统在产生一个线程，或是在各个线程之间作切换工作时，负担要比进程小得多，也正因为如此，线程也被称为**轻量级进程**。

## ● 开源软件特训营总结

今日通过 Git 将远程仓库中的一个文件进行了删除，熟悉了相关的 Git 操作；  
同时对远程仓库中的文件进行了整理，重新建立了文件夹将文件放下相应目录下。