

# 修理牧场

## 1.项目简介

农夫要修理牧场的一段栅栏，他测量了栅栏，发现需要N块木头，每块木头长度为整数 $L_i$ 个长度单位，于是他购买了一个很长的，能锯成N块的木头，即该木头的长度是 $L_i$ 的总和。但是农夫自己没有锯子，请人锯木的酬金跟这段木头的长度成正比。为简单起见，不妨就设酬金等于所锯木头的长度。例如，要将长度为20的木头锯成长度为8，7和5的三段，第一次锯木头将木头锯成12和8，花费20；第二次锯木头将长度为12的木头锯成7和5花费12，总花费32元。如果第一次将木头锯成15和5，则第二次将木头锯成7和8，那么总的花费是35（大于32）

## 2.项目功能要求

**输入格式：**输入第一行给出正整数N（ $N < 10^4$ ），表示要将木头锯成N块。第二行给出N个正整数，表示每块木头的长度。

**输出格式：**输出一个整数，即将木头锯成N块的最小花费。

## 3.算法思想

本题采用贪心算法的思想。首先将分解问题该为思考求和问题，即将问题转化成：已知n块木块长度如何按照求其合成最小值问题。这道题和NOIP2014年提高组的[合并果子](#)的思考方式一模一样：总是取数值较少的两个进行合并，并将合并后的数值并入数组中，直到全部完全合成。

## 4.核心代码及功能

- 预置变量的声明

```
int wood[10000];           //用于保存所有的木头长度，默认最大值为10000
int N;                     //保存共有多少个木头
long long answer=0;        //保存合成所有木块的花费总和
```

- 读入时非法输入的处理

这里的非法输入，即输入的木头长度小于等于0时，所以当遇到这种情况时我们将计数器减一重新读入该位木头长度。

```
for(int i=0;i<N;i++){
    cin>>wood[i];
    if(wood[i]<=0){           //用于检测输入是否存在非法，若非法则重新输入该数据
        cout<<"the length of wood can not be negative!!"
            "please reinput this length:";
        i--;
        continue;
    }
}
```

- 算法表示

这里取最小的两个合并后我们没有重新sort整个数组，而是用插入排序的方式将合成的数插入到数组中。因为此时大部分数组都已经是有序的了，这时再使用快速排序的方式是划不来的。

```
sort(wood,wood+N);          //按照从小到大的顺序给数组排序
for(int i=1;i<N;i++){
    wood[i]+=wood[i-1];      //每次取最小的两个木头进行合成
    answer+=wood[i];
    for(int j=i;j<N-1;j++){  //去除之前的两个木头并将合成后的木头放入数组中
```

```
        if(wood[j]<wood[j+1]){
            swap(wood[j],wood[j+1]);
        }
        else{
            break;
        }
    }
}
cout<<answer<<endl;
```

## 5.项目实例

---

```
/Users/kirito/CLionProjects/untitled/cmake-build-debug/untitled
please input N:3
please input the number of N woods:
5 7 8
32

Process finished with exit code 0
```