

# 大作业

## 回溯剪支法

对背包数据进行从小到大排序得到：

{10 40} {25 30} {30 40} {35 10} {40 35} {50 50} {60 30}

排序有利于在回溯过程中剪支

当回溯进行到深度*i*的时候，如果当前的重量 $tw$ +背包重量 $weight[i]>$ 最大重量 $W$ ,就无需进行左右分支搜索

In[118]:=

```
(tw+weight[i]) ≤ W
```

当回溯进行到深度*i*的时候，如果当前的价值 $tv$ +剩余背包的价值 $rv<$ 目前的最大价值 $maxv$ ,就无需进行分支搜索.

In[119]:=

```
tv+rv < maxv
```

当回溯进行到深度*i*的时候，如果当前的重量 $tw$ +剩余的背包重量 $rw\leq$ 最大重量 $W$ ，就无需进行右分支搜索。

In[120]:=

```
tw+rw ≤ W
```

## 运行结果

排序后的背包{重量 价值}数组:{10 40} {25 30} {30 40} {35 10} {40 35} {50 50} {60 30}

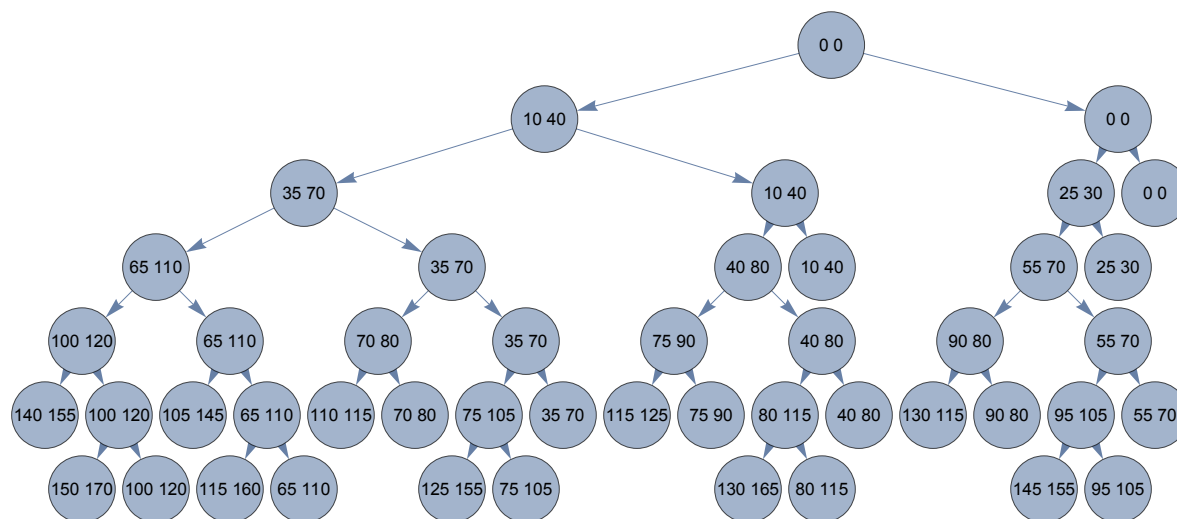
最大价值:170

取得最大价值时候的背包重量： 150

取得最大价值的【0-1】数组:0101111

## 状态空间搜索树

Out[121]=



## 代码与实现环境

编程语言: C++

编译器: Apple clang version 14.0.0 (clang-1400.0.29.202)

语言标准: C++20

操作系统: macOS13

## 使用makefile编译

### Mathematica画出树形图