Algorithm HW7

1.

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 20 & 30 & 10 & 11 \\ 2 & 20 & \infty & 16 & 4 & 2 \\ 3 & 30 & 16 & \infty & 6 & 7 \\ 4 & 10 & 4 & 6 & \infty & 12 \\ 5 & 11 & 2 & 7 & 12 & \infty \end{array}$$

1: 10
2: 2
3: 6
4: 4
5: 2

步骤1: 根节点A: 计算初始下界: $V1:(10,11)\to 21$, $V2:(2,4)\to 6$

$V3:(6,7)\to 13$, $V4:(4,6)\to 10$, $V5:(2,7)\to 9$

$LB(A)=\lceil (21+6+13+10+9)/2\rceil=\lceil 59/2\rceil=30$

步骤2: 从V1分枝: $1\to 2$: $\hat{c}(1,2)=\left\lceil\frac{(20+10)+(20+2)+13+10+9}{2}\right\rceil=42$.

$1\to 3$: $\hat{c}(1,3)=\left\lceil\frac{(30+10)+(30+6)+6+10+9}{2}\right\rceil=51$

$1\to 4$: $\hat{c}(1,4)=\left\lceil\frac{(10+10)+(10+4)+6+13+9}{2}\right\rceil=$ 2 取该

$1\to 5$: $\hat{c}(1,5)=\left\lceil\frac{(11+10)+(11+2)+6+13+10}{2}\right\rceil=32$

~~步骤~~3: $1\to 4\to x$: $\hat{c}(1,4,2)=\left\lceil\frac{\cancel{10+4+11+2+10+4+13+9}}{2}\right\rceil=30$.

$\hat{c}(1,4,3)=\left\lceil\frac{(10+11)+(10+6)+(6+7)+6+9}{2}\right\rceil=30$.

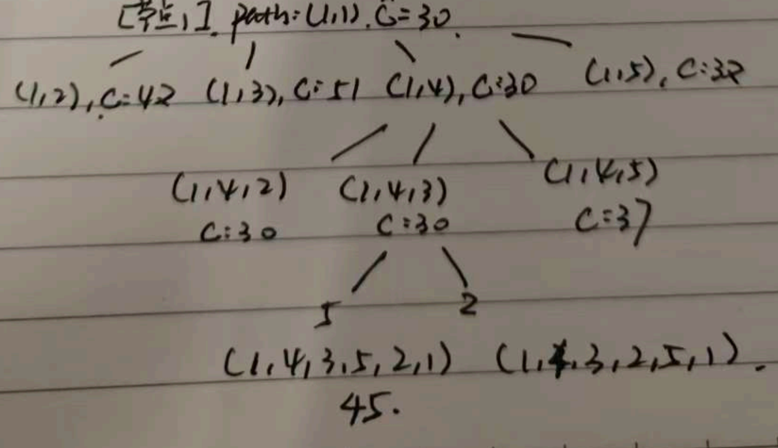
$\hat{c}(1,4,5)=\left\lceil\frac{(10+11)+(12+2)+(10+12)+6+13}{2}\right\rceil=37$.

Step4: ~~$1\to 4\to 2\to x$: $\hat{c}(1,4,2,3)=$~~ $\left\lceil\frac{(10+11)+(10+4)+(2+16)+(16+6)+9}{2}\right\rceil$

$1\to 4\to 2\to 5\to 3\to 1$: $10+4+2+7+30=53$.

$1\to 4\to 2\to 3\to 5\to 1$: $10+4+2+16+7+11=48$. 更新上界 $u=48$.

剪枝后, $1\to 4\to 3\to 5\to 2\to 1$ 45.



2. 伪代码：

```
Sort items by decreasing v[i]/w[i]
Initialize priority queue Q
bestValue = 0

root:
    u = 0, cw = 0, cv = 0
    UB = Bound(0, 0, 0)
Push root into Q

while Q not empty:
    x = pop node with max UB
    if x.UB <= bestValue:
        continue   // 剪枝

    if x.u == n:
        bestValue = max(bestValue, x.cv)
        continue

    // 左子结点：选择第 u+1 件物品
    if x.cw + w[u+1] <= C:
        y.cw = x.cw + w[u+1]
        y.cv = x.cv + v[u+1]
        y.u = x.u + 1
        y.UB = Bound(y)
        if y.cv > bestValue:
            bestValue = y.cv
        Push y into Q

    // 右子结点：不选第 u+1 件物品
    z.cw = x.cw
    z.cv = x.cv
    z.u = x.u + 1
    z.UB = Bound(z)
    if z.UB > bestValue:
        Push z into Q
```

实例：背包容量为 50。

物品 i	重量 w_i	价值 v_i	v_i/w_i
1	2	40	20
2	3	50	16.7
3	5	60	12
4	4	40	10
5	6	54	9
6	7	56	8
7	8	56	7
8	9	63	7
9	10	60	6
10	11	66	6
11	12	60	5
12	13	65	5
13	14	70	5
14	15	75	5
15	16	64	4
16	18	72	4

根结点上界 按分数背包装入：装满前 8 件后剩余容量，用第 9 件分数补足：

$$UB_{root} \approx 40+50+60+40+54+56+56+63+\frac{(50-44)}{10}\times 60 = 419$$

搜索过程 使用最大上界优先扩展，每次分为：左子结点（选当前物品），右子结点（不选当前物品）。当结点：

$$UB \le \text{当前最优值}$$

时直接剪枝。经过优先队列搜索与剪枝，最终得到最优装法。

最终结果 最优选择物品集合：$\{1,2,3,4,5,6,7,8\}$

总重量：$2+3+5+4+6+7+8+9=44\le 50$

总价值：$40+50+60+40+54+56+56+63=\boxed{419}$

3.

形式化：

- n 个任务：$J = 1, \dots, n$
- k 台完全相同的并行机器
- 任务 i 的处理时间为 t_i
- 每个任务不可拆分，每次只能在一台机器上执行
- 目标：最小化

$$C_{\max} = \max_{j=1}^{k} \text{第 } j \text{ 台机器的总负载}$$

这是经典问题：$P||C_{\max}$

分枝限界算法的状态空间建模：

一个结点表示为：$X = (u, L_1, L_2, \dots, L_k)$

含义：

- u：已经分配了前 u 个任务（任务按固定顺序）
- L_j：第 j 台机器当前负载

在结点 X，对第 $u+1$ 个任务：

- 产生最多 k 个子结点
- 将任务 $u+1$ 分配给任意一台机器

这是**标准分枝**，不是贪心。

下界：对任意结点 X：

$$LB(X) = \max\left(\max_j L_j,;\frac{\sum_{i=1}^{n} t_i}{k}\right)$$

含义：已有最大负载；平均负载下界。

上界：LPT（Longest Processing Time）算法：

- 将**未分配任务**按处理时间递减排序
- 依次把任务分配给**当前负载最小的机器**

得到一个完整调度，其完工时间：$UB(X)$

```
Sort jobs in non-increasing order of t_i
Compute initial upper bound C* using LPT

Priority Queue Q
Push root node X0 = (u=0, all Lj=0) into Q

while Q not empty:
    X = pop node with smallest LB

    if LB(X) >= C*:
        continue   // 剪枝

    if X.u == n:
        C* = min(C*, max_j Lj)
        continue

    for j = 1 to k:
        create child X':
            X'.u = X.u + 1
            X'.Lj = X.Lj + t_{u+1}
            X'.Lothers = X.Lothers

        compute LB(X')
        compute UB(X') using LPT

        if UB(X') < C*:
            C* = UB(X')

        if LB(X') < C*:
            push X' into Q
```