《算法分析与设计》期末试卷参考答案（一）

计算机科学与技术2019

**一、单选题（每题2分，共40分）**

B C B B C C B C D B

D D B C A D A D C D

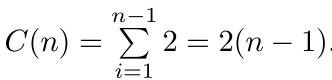
**二、综合题（共60分）**

**1、（10分）解：**

（1）算法功能：计算数组最大值与最小值之差，即元素值的范围。 **（1分）**

基本操作：（每次循环都要执行）2次元素值比较A[i] < minval和A[i] > maxval，则取这两次比较为基本操作 **（2分）**

（2）令C(n)是基本操作的执行次数，则计算公式如下：

 **（2分）**

迭代求解：C(n)=2(n-1) **（1分）**

（3）渐近效率：C(n) = Θ(n) **（2分）**

（4）当A[i] < minval时，就无需再执行A[i] > maxval，所以可有如下改进：

if A[i] < minval minval 🡨 A[i]

else if A[i] > maxval maxval 🡨 A[i] **（2分）**

**评分标准：功能描述、基本操作识别、基本操作执行次数公式推导及其求解、渐近效率分析、改进措施描述完整、正确或等价得满分。部分正确，酌情扣分。**

**2、（10分）解：**

（1）**分析**：令时间复杂度＝T(n)：输入规模=n时，执行的指令总条数；运算速度＝S：单位时间内执行的指令条数。

输入规模=n，算法执行时间T1＝T(n)/S；输入规模增长m倍=m×n，算法执行时间T2＝T(m×n)/S。故，执行时间倍数=T2/T1=T(m×n)/T(n) （3分）

**计算**：T(1000)/T(500)=(1/2\*10002)/(1/2\*5002)=(2\*500)2/5002=22=4，即在输入规模增长2倍的情况下，算法运行时间将多4倍。 （2分）

**评分标准：变量定义、关系式推导、计算正确或等价得满分。部分正确，酌情扣分。**

（2）**分析**：令当前输入规模＝n1的时间复杂度＝T(n1)，设执行时间＝t、令计算速度=s1，则t=T(n1)/s1；当速度增长m倍（s2=m×s1），执行时间不变=t，设输入规模=n2，则t=T(n2)/s2成立。综合则有T(n1)/s1=t=T(n2)/s2=T(n2)/(m×s1)，整理得：

T(n2)＝mT(n1)，分解关系式，就能算出规模增长倍数。 （3分）

**计算**：T(n1)=1/2\*n12，T(n2)=(1/2)\*n22，由上有T(n2)=100T(n1)

即(1/2)\*n22=100\*(1/2)\*n12=>(n2/n1)2=100=>n2/n1=10，即在速度提升100倍的情况下，算法在相同时间内处理的输入规模多10倍。 （2分）

**评分标准：变量定义、关系式推导、计算正确或等价得满分。部分正确，酌情扣分。**

**3、（20分）解：**

（1）① 硬币按面值从大到小排序为11分、5分、1分 （1分）

② 按面值从大到小扫描硬币，进行兑换：15 = 1 ×11 + 4 = 1 ×11 + 0 × 5 + 4

= 1 ×11 + 0 × 5 + 4 × 1，即用{ 1枚11分，4枚1分}共5枚完成兑换 （2分）

③ 贪心求解结果不是最优，∵15=5×3，即3枚5分是更优解 （2分）

**评分标准：面值排序、兑换过程、最终结果、判断理由完整、正确或等价得满分。部分正确，酌情扣分。**

（2）**动态规划**

**法1：**

① 定义面值数组d[1], d[2], …, d[m]=1，兑换金额=n （1分）

② 定义最优值函数=c (i, j)：可选面值为d[i], d[i +1], ……, d[m]，兑换金额=j

其中，1≤i≤m，0≤j≤n （2分）

③ 最优值的递归计算公式为：

当d[i] ＞ j 时，c (i, j) = c ( i+1, j )

当d[i] ≤ j 时，c (i, j) = min{ c(i+1, j), j / d[i] + c(i, j mod d[i]) }

初始条件：c ( m, j ) = j（∵d[m]=1） （3分）

④ 实例：d[1]=11, d[2]=5, d[3]=1，n=15

最优值矩阵计算结果： （4分）



⑤ 最优值=c(1, 15)=3（即兑换15分最少需要3枚硬币） （1分）

⑥ 最优解构造：

c(1,15)=3=c(2,15)，即11分硬币无需参与兑换，否则应有c(1,15)≠c(2,15)

c(2,15)=3≠15=c(3,15)，即5分硬币要参与兑换，兑换方案是：15 = 3×5 +0

c(2, 0)=0=c(3, 0)，即5分硬币不再使用，1分硬币也无需使用

c ( 2, 15 mod 5) = 3+c(2,0)= 3 < 15=c(1,15)，则1未用 （3分）

⑦ 最优解= { 0枚11分，3枚5分，0枚1分} （1分）

**法2：**① 定义面值数组d[1]=1, d[2], …, d[m]，兑换金额=n （1分）

② 最优值函数F (n)：兑换金额n所需的最少硬币数 （2分）

③ 最优值的递归计算公式为：

n=0，F (n) = 0

n>0，F (n) = min{ F( n- dj) } + 1，其中j满足n≥dj （3分）

④ 实例：d[1]=1, d[2]=5, d[3]=11，n=15

最优值矩阵计算结果： （4分）

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| n | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| F | 0 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 5 | 2 | 1 | 2 | 3 | 4 | 3 |

⑤ 最优值=F(15)=3（即兑换15分最少需要3枚硬币） （1分）

⑥ 最优解构造：

∵F(15)=3由F(10)+d[2]得到∴用了1枚5分硬币，剩余金额=10

∵F(10)=2由F(5)+d[2]得到∴用了1枚5分硬币，剩余金额=5

∵F(5)=1有F(0)+d[2]得到∴用了1枚5分硬币，完成兑换 （3分）

⑦ 最优解= { 0枚11分，3枚5分，0枚1分} （1分）

**评分标准：最优值函数定义、实例阐述、最优值（矩阵）计算、最优解构造过程完整、正确或等价得满分。部分正确，酌情扣分。**

**4、（20分）解：**

搜索技术2种：回溯、分支定界；0-1背包可以定义3种价值上界函数（剩余物品全部装包、剩余物品连续装包、剩余物品最大单价装包）；分支定界搜索中，活结点可以有普通和优先2种队列。以下为部分参考解答。

**参考1**：① 分支定界搜索技术 （2分）

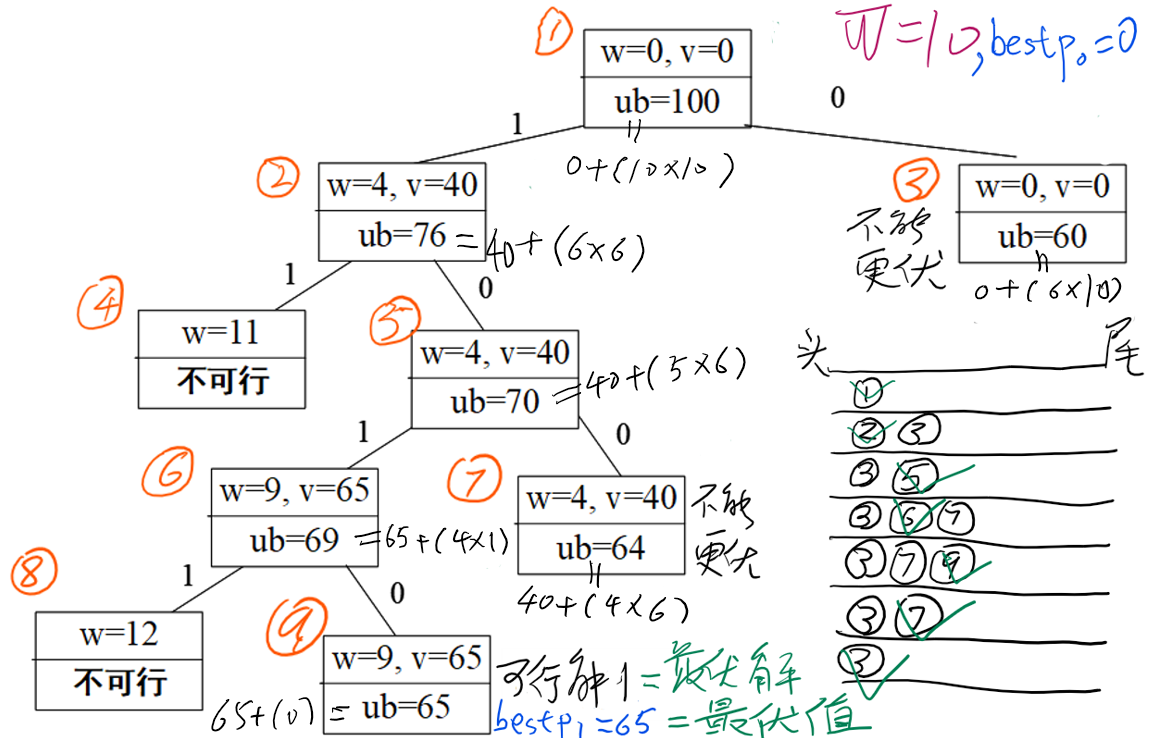
② 物品按单位价值从高到低排序 （2分）

|  |  |  |  |
| --- | --- | --- | --- |
| **物品** | **重量** | **价值** | **单价** |
| 1 | 4 | 40 | 10 |
| 2 | 7 | 42 | 6 |
| 3 | 5 | 25 | 5 |
| 4 | 3 | 12 | 4 |

③ 价值上界＝当前包内总价值＋剩余物品最大单价\*剩余容量 （2分）

④ 采用优先队列，价值上界大结点优先出队 （2分）

⑤ 分支定界搜索生成的状态空间树 （10分）



图中：W：背包容量，bestp：当前状态下的装包最大价值；w：包内物品总重量，v：包内物品总价值；ub：当前状态下，可能的装包总价值上界

图中：红色编号表示了结点的生成顺序；结点生成时完成装包重量w和价值v、上界ub的计算；通过约束、上界检查的成为活结点，进入队列；绿勾结点有最大上界，被选中出队。

⑥ 状态空间树中，从根结点到最优解结点路径上的标记排列出来，就是最优解向量=(1, 0, 1, 0)，故最优装包方案＝{ 物品1，物品3 } （2分）

**参考2**：① 分支定界搜索技术 （2分）

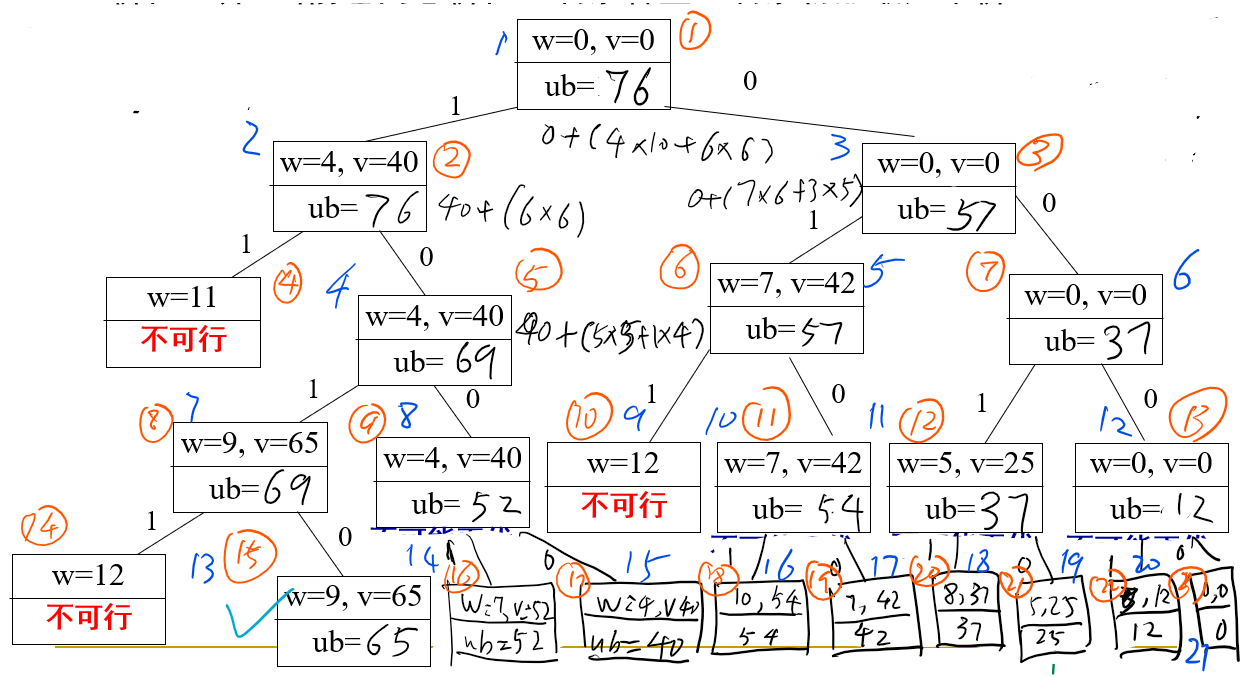
② 物品按单位价值从高到低排序 （2分）

|  |  |  |  |
| --- | --- | --- | --- |
| **物品** | **重量** | **价值** | **单价** |
| 1 | 4 | 40 | 10 |
| 2 | 7 | 42 | 6 |
| 3 | 5 | 25 | 5 |
| 4 | 3 | 12 | 4 |

③ 价值上界＝当前包内总价值＋剩余物品连续装包 （2分）

④ 采用普通队列，先进先出 （2分）

⑤ 分支定界搜索生成的状态空间树 （10分）



图中：W：背包容量，bestp：当前状态装包最大价值；w：包内物品总重量，v：包内物品总价值；ub：当前状态下，装包总价值上界

图中：红色编号表示了结点的生成顺序；结点生成时完成装包重量w和价值v、上界ub的计算；通过约束、上界检查的成为活结点，进入队列；队头结点出队，蓝色编号为出队顺序。

⑥ 状态空间树中，从根结点到最优解结点路径上的标记排列出来，就是最优解向量=(1, 0, 1, 0)，故最优装包方案＝{ 物品1，物品3 } （2分）

**参考3**：① 回溯搜索、界限函数加速 （2分）

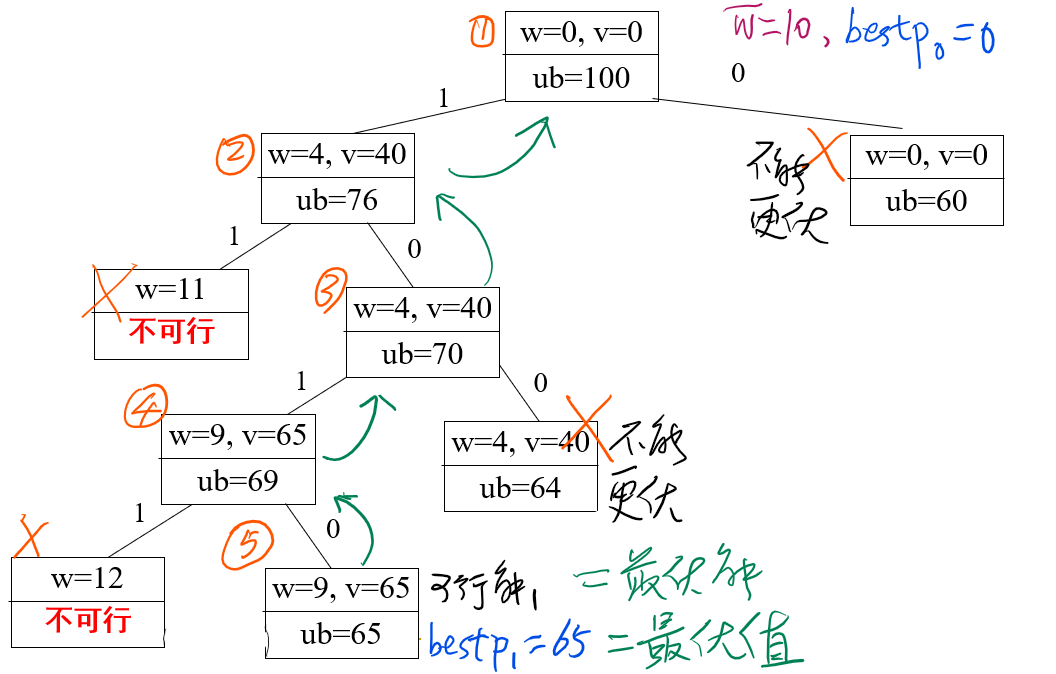
② 物品按单位价值从高到低排序 （2分）

|  |  |  |  |
| --- | --- | --- | --- |
| **物品** | **重量** | **价值** | **单价** |
| 1 | 4 | 40 | 10 |
| 2 | 7 | 42 | 6 |
| 3 | 5 | 25 | 5 |
| 4 | 3 | 12 | 4 |

③ 价值上界＝当前包内总价值＋剩余物品最大单价\*剩余容量 （2分）

④ 搜索进行中，不满足约束者、价值上界不能更优者被提前剪枝 （2分）

⑤ 回溯搜索生成的状态空间树 （10分）



图中：W：背包容量，bestp：当前状态的装包最大价值；w：包内物品总重量，v：包内物品总价值；ub：当前状态下，装包总价值上界

图中：红色编号表示了结点被搜索的顺序；生成结点时，完成装包重量w和价值v的试算，完成上界ub的计算；通过约束、上界检查的结点，立即进入搜索，未通过检查的结点被剪枝（放弃）；绿箭头表示回溯方向。

⑥ 状态空间树中，从根结点到最优解结点路径上的标记排列出来，就是最优解向量=(1, 0, 1, 0)，故最优装包方案＝{ 物品1，物品3 } （2分）

**评分标准：搜索技术选择合理、加速方法选择理由合理，价值上界函数定义及描述清晰，状态空间树的构造过程（如价值上界、工作变量、搜索顺序等）描述完整、计算正确得满分。部分正确，酌情扣分。**