HW4

范潇　2254298

2024 年 5 月 16 日

题目 1. (4-1) 按 $l_1 \geq l_2 \geq \cdots \geq l_n$ 的次序来考虑程序时的一个反例输入为：10,2,6,6。

如果按照贪心算法, 会将 10 分配给 A, 然后将 2, 6, 6 都分配给 B。此时 $\max\{\sum_{i\in A} l_i, \sum_{i\in B} l_i\} = 14$。但是最优解之一为 $A = \{10, 2\}, B = \{6, 6\}$，使得 $\max\{\sum_{i\in A} l_i, \sum_{i\in B} l_i\} = 12$。

按 $l_1 \leq l_2 \leq \cdots \leq l_n$ 的次序来考虑程序时的一个反例输入为：1,1,2。

如果按照贪心算法,会将 1 分配给 A,然后将 1 都分配给 B,最后将 2 分配给 A。此时 $\max\{\sum_{i\in A} l_i, \sum_{i\in B} l_i\} = 3$。但是最优解之一为 $A = \{1, 1\}, B = \{2\}$，使得 $\max\{\sum_{i\in A} l_i, \sum_{i\in B} l_i\} = 2$。

显然, 对于给定输入, $\sum_{1\leq i\leq n} l_i$ 是定值, 又由于对称性, 不失一般性, 可以只考虑使得 $\sum_{i\in A} l_i \leq \sum_{i\in B} l_i$ 的解。此时，$\sum_{1\leq i\leq n} l_i = \sum_{i\in A} l_i + \sum_{i\in B} l_i \geq 2\sum_{1\in A} l_i$，不妨设 l_i 为整数，记 $\sum_{1\leq i\leq n} l_i = M$ 则

$$\sum_{i\in A} l_i \leq \lfloor \frac{M}{2} \rfloor$$

又因为

$$\max\{\sum_{i\in A} l_i, \sum_{i\in B} l_i\} = \max\{\sum_{i\in A} l_i, M - \sum_{i\in A} l_i\} = M - \sum_{i\in A} l_i$$

所以原问题等价于，求 $A \subseteq \{1, 2, \cdots, n\}$

$$\text{minimize} \quad M - \sum_{i\in A} l_i$$

$$\text{subject to} \quad \sum_{i\in A} l_i \leq \lfloor \frac{M}{2} \rfloor, l_i \in \mathbb{Z}^+$$

也等价于

$$\text{maximize} \quad \sum_{i\in A} l_i$$

$$\text{subject to} \quad \sum_{i\in A} l_i \leq \lfloor \frac{M}{2} \rfloor, l_i \in \mathbb{Z}^+$$

将问题“使得 $\sum_{i\in A} l_i \leq x$ 且 $\sum_{i\in A} l_i$ 尽可能地大 $(A \subseteq \{1, 2, \cdots, y\})$”的最优解对应的 $\sum_{i\in A} l_i$ 记为 $m[x, y]$，则有

$$m[i,j] = \begin{cases} m[i, j-1] & v_j > i \\ \max\{m[i - v_j, j-1] + v_j, m[i, j-1]\} & else \end{cases}$$

显然, 当 $l_j > i$ 时, m[i,j] 中不可能包含 l_j, 因为这违反了不等式约束, 从而 m[i,j] 与 m[i,j-1] 对应的合法的解相同, 最优解也相同, 从而 m[i,j]=m[i,j-1]。当 $l_j \leq i$ 时, 由反证法易得, 若 $j \notin A$, 则 $A \subseteq \{1, 2, \cdots, j-1\}$,

对应的最优解为 $m[i,j-1]$ 的最优解；若 $j\in A$，则对应的最优解为 $\{j\}\cup A'$，其中 A' 对应 $m[i-l_j,j-1]$ 的最优解。

原问题的最优解对应的和便是 $m[\lfloor\frac{M}{2}\rfloor,n]$。想要计算得到 $m[\lfloor\frac{M}{2}\rfloor,n]$，共需计算 $\lfloor\frac{M}{2}\rfloor\cdot n$ 个值，所以时间复杂度为 $\Theta(nM)$。

为了回溯还原最优解,需要保存过程中计算得到的 $m[i,j]$ 值。回溯过程从 $m[\lfloor\frac{M}{2}\rfloor,n]$ 开始,若 $m[x,y]==m[x,y-1]$, 则说明 $m[x,y]$ 的最优解的构成和 $m[x,y-1]$ 相同。否则, 说明 $m[x,y]$ 对应的最优解中, $l_y\in A$, 剩余部分由是 $m[x-l_y,y-1]$ 对应的最优解。回溯过程所需要的时间复杂度为 $\Theta(M+n)$

当 $j=0$ 时，说明 $m[i,j]$ 对应的最优解中 $A=\varnothing$，所以 $m[i,0]=0$。当 $i=0$ 时，由于 $v_k>0$，最优解中 $A=\varnothing$，从而 $m[0,j]=0$。

Algorithm 1: DynamicProgrammingApproach(values)

```
// values 下标从 1 开始
M = ⌊sum (values)/2⌋
l = len (values)
初始化一个 (M+1)×(l+1) 的全零数组 dp
for i in [1..M] do
    for j in [1..l] do
        if values[j]>i then
            dp[i][j] = dp[i][j-1]
        else
            dp[i][j] = max(dp[i][j-1],dp[i-values[j]][j-1]+values[j])
        end if
    end for
end for
A=[]// 开始回溯
x = M
y = l
while x and y do
    while y and dp[x][y] == dp[x][y-1] do y -= 1
    if y==0 then break
    A.append(y)
    x -= values[y-1]
    y -= 1
end while
return A
```