1.蓝桥课程抢购 - 蓝桥云课 (lanqiao.cn)

语言整理

- 对每一项，如果能在[0, B]开始执行，花费A时间，可以获得C钱
- 最多获得多少钱?

解题思路

0-1背包问题 / “选或不选”问题

$$\text{定义}mx = \max(B), \text{由于所有项一定在}[0, mx]\text{内开始执行，按照}B\text{从小到大排序遍历。}$$
$$\text{定义}d[i][j]\text{为考察前}i\text{项，恰好花}j\text{时间内获得最多的钱}$$
$$\text{如果第}i\text{项执行，}d[i][j] = max(d[i-1][j-b] + c, d[-1i][j])$$
$$\text{注意需要判断}j\text{的界限是在}[a, b]\text{内}$$
$$d\text{的维度，}(n+1)\cdot(mx+1)$$

时间复杂度：$O(n \times \max(B))$

```python
# 语言整理：
# 对每一项，如果能在[0, B]开始执行，花费A时间，可以获得C钱
# 最多获得多少钱？
import sys
input = lambda: sys.stdin.readline().strip()

n = int(input())
nums = []
for _ in range(n):
    a, b, c = map(int, input().split())
    nums.append((a, b, c))
# 按照B排序
nums.sort(key = lambda x: x[1])
# 所有项一定在[0, mx]内开始执行
mx = nums[-1][1]


d = [[0] * (mx + 1) for _ in range(n + 1)]
for i in range(1, n + 1):
    a, b, c = nums[i - 1]
    for j in range(1, mx + 1):
        if a <= j <= b: # “选”
            d[i][j] = max(d[i - 1][j - a] + c, d[i - 1][j])
        else:
            d[i][j] = d[i - 1][j]
print(max(d[n]))
```