

### 算法实现题 3-2 最优批处理问题

#### ★问题描述:

在一台超级计算机上, 编号为 $1, 2, \dots, n$ 的 $n$ 个作业等待批处理。批处理的任务就是将这 $n$ 个作业分成若干批, 每批包含相邻的若干作业。从时刻 $0$ 开始, 分批加工这些作业。在每批作业开始前, 机器需要启动时间 $S$ , 而完成这批作业所需的时间是单独完成批中各个作业需要时间的总和。单独完成第 $i$ 个作业所需的时间是 $t_i$ , 所需的费用是它的完成时刻乘以一个费用系数 $f_i$ 。同一批作业将在同一时刻完成。例如, 如果在时刻 $T$ 开始一批作业 $x, x+1, \dots, x+k$ , 则这一批作业的完成时刻均为 $T + S + (t_x + t_{x+1} + \dots + t_{x+k})$ 。最优批处理问题就是要确定总费用最小的批处理方案。例如, 假定有 $5$ 个作业等待批处理, 且

$$S = 1, (t_1, t_2, t_3, t_4, t_5) = (1, 3, 4, 2, 1), (f_1, f_2, f_3, f_4, f_5) = (3, 2, 3, 3, 4)$$

如果采用批处理方案 $\{1, 2\}, \{3\}, \{4, 5\}$ , 则各作业的完成时间分别为 $(5, 5, 10, 14, 14)$ , 各作业的费用分别为 $(15, 10, 30, 42, 56)$ , 因此, 这个批处理方案总费用是 $153$ 。

#### ★算法设计:

对于给定的待批处理的 $n$ 个作业, 计算其总费用最小的批处理方案。

#### ★数据输入:

由文件 input.txt 提供输入数据。文件的第 $1$ 行是待批处理的作业数 $n$ , 第 $2$ 行是启动时间 $S$ 。接下来每行有 $2$ 个数, 分别为单独完成第 $i$ 个作业所需的时间是 $t_i$ 和所需的费用系数 $f_i$ 。

#### ★结果输出:

将计算出的最小总费用输出到文件 output.txt 中。

#### 输入文件示例

input.txt

5

1

1 3

3 2

4 3

2 3

1 4

#### 输出文件示例

output.txt

153