1. 定义子问题

令 $\text{OPT}(i)$ 表示从第 i 天到第 n 天期间（包括第 i 天），满足所有汽油需求并最小化总成本的最优解。我们假设在第 $i-1$ 天结束时储罐是空的，并且在第 i 天需要进行一次订单。我们的最终目标是计算 $\text{OPT}(1)$。

2. 递推关系

为了计算 $\text{OPT}(i)$，我们考虑在第 i 天下一笔订单。这笔订单将满足从第 i 天到第 $j-1$ 天的需求（其中 j 是下一次订单的日期，或者 $j=n+1$ 表示这次订单将满足直到第 n 天的所有剩余需求）。该策略的总成本包括：

1. 运费：P (因为在第 i 天下了一个订单)。
2. 存储费：这批在第 i 天到达的汽油，将满足从第 i 天到第 $j-1$ 天的需求。对于第 k 天的需求 g_k，它在储罐中被存储了 $k-i$ 天。因此，这部分的总存储费用是 $\sum_{k=i}^{j-1} g_k \cdot (k-i) \cdot c$。
3. 后续子问题的最优成本：从第 j 天开始到第 n 天的最优成本，即 $\text{OPT}(j)$。

因此，递推关系为：

$$\text{OPT}(i) = \min_{i<j\leq n+1} \left\{ P + \left(\sum_{k=i}^{j-1} g_k \cdot (k-i) \cdot c \right) + \text{OPT}(j) \right\}$$

其中，需要满足容量限制：在任何时候，储罐中的油量都不能超过 L。这意味着对于第 i 天下的订单，其满足的量 $\sum_{k=i}^{j-1} g_k$ 必须 $\leq L$。

3. 边界条件

我们定义 $\text{OPT}(n+1)=0$，表示在第 $n+1$ 天（休业后）没有额外的成本。

4. 计算顺序

我们从 $i=n$ 倒序计算到 $i=1$。

- 对于 $i=n$：$\text{OPT}(n) = \min_{n<j\leq n+1} \left\{ P + \left(\sum_{k=n}^{j-1} g_k \cdot (k-n) \cdot c \right) + \text{OPT}(j) \right\}$ 唯一可能的 j 是 $n+1$。$\text{OPT}(n) = P + (g_n \cdot (n-n) \cdot c) + \text{OPT}(n+1) = P+0+0 = P$。(此处的容量限制是 $g_n \leq L$)

5. 恢复最优策略

在计算每个 $\text{OPT}(i)$ 时，我们记录下使最小值达到的那个 j 值（即下一次订货的日期）。通过从 $\text{OPT}(1)$ 开始回溯这些记录的 j 值，我们可以确定所有订单的日期和每次的订货量。

6. 运行时间分析

我们需要计算 n 个 $\text{OPT}(i)$ 值（从 $i=n$ 到 $i=1$）。对于每个 $\text{OPT}(i)$ 的计算，我们需要遍历 j，大约有 $n-i$ 个选项。内层循环中的求和项 $\sum_{k=i}^{j-1} g_k \cdot (k-i) \cdot c$ 可以通过预计算或者在循环中递增维护的方式进行优化，使其计算时间为 $O(1)$ 或 $O(j-i)$。

如果对 $\sum_{k=i}^{j-1} g_k \cdot (k-i)$ 和 $\sum_{k=i}^{j-1} g_k$ 进行预处理（计算前缀和），那么计算每个 j 选项的成本将是 $O(1)$。* 令 $G_s[x] = \sum_{k=1}^{x} g_k$ (需求量前缀和) * 令 $D_c[x] = \sum_{k=1}^{x} g_k \cdot k$ (需求量乘以天数的前

缀和) * 则 $\sum_{k=i}^{j-1} g_k = G_s[j-1] - G_s[i-1]$ * 而 $\sum_{k=i}^{j-1} g_k \cdot (k-i) = \sum_{k=i}^{j-1} g_k \cdot k - i \cdot \sum_{k=i}^{j-1} g_k = (D_c[j-1] - D_c[i-1]) - i \cdot (G_s[j-1] - G_s[i-1])$ 这两个求和可以在 $O(1)$ 时间内计算。

因此，计算每个 $\mathrm{OPT}(i)$ 的时间复杂度是 $O(n-i)$。总的运行时间复杂度为 $\sum_{i=1}^{n} O(n-i) = O(n^2)$。这与原题所说的 $O(n^2)$ 时间复杂度相符。