

КОНТРОЛНО СЪСТЕЗАНИЕ ЗА РАЗШИРЕН

МОМИЧЕШКИ ОТБОР 27 ЮНИ 2022

Група G

Анализ на задача Опашка

Тази задача беше вдъхновена по истински случай, докато чаках пред личната лекарка :)

Първа подзадача. Тъй като опашката е само от двама човека е достатъчно да проверим коя от двете възможни наредби е оптимална. Тук ще е добре да отбележим точната сметка, която правим: ако $\max(w_1, w_2 + t_1) \leq \max(w_2, w_1 + t_2)$ то наредбата (1,2) е не по-лоша от наредбата (2,1).

Втора подзадача. Идеята на тази подзадача беше да подтикне разсъждения в посока някакъв вид сортиране на хората вместо просто 3! брут-форс (не особено сполучливо).

Трета подзадача. Тук идва основното наблюдение в задачата, което ни помага да намерим оптималната подредба на всички клиенти. Решението е вид грийди сортиране, което се среща и в други задачи. Общия модел на такива решения винаги е следният:

1. Решаваме задачата за 2 елемента и съставяме оператор за сравнение
2. Сортиране всички елементи с помощта на оператора

Нека бележим с $P_i \leq P_j$, когато $\max(w_i, w_j + t_i) \leq \max(w_j, w_i + t_j)$, или с други думи, когато е по-оптимално клиент P_i да бъде обслужен преди клиент P_j , ако теоретично само те са на опашката.

Ще покажем две твърдения за $\leq$, които естествено водят до грийди сортирането:

При дадена наредба на клиентите $P_1, P_2, \dots, P_i, P_{i+1}, \dots, P_N$, ако $P_{i+1} \leq P_i$, то разменяйки позициите им ще получим не по-лоша наредба. Това лесно може да се докаже, като отбележим, че времето на чакане на всички останали не се променят, само на P_i и P_{i+1} .

Операторът $\leq$ има свойството, което ни е необходимо за еднозначно сортиране, а именно транзитивност: за i, j, k ако $P_i \leq P_j$ и $P_j \leq P_k$, то $P_i \leq P_k$. Това може да се докаже емпирично с програма, ако искаме да тестваме подобна идея. Реалното доказателство е просто алгебра:

Доказателство:

Дадено: $\max(w_i, w_j + t_i) \leq \max(w_j, w_i + t_j)$, $\max(w_j, w_k + t_j) \leq \max(w_k, w_j + t_k)$

- Случай 1: $w_j \leq w_i + t_j \Rightarrow w_i + t_j \geq w_j + t_i$
 - Случай 1.1: $w_k \leq w_j + t_k \Rightarrow w_j + t_k \geq w_k + t_j \Rightarrow w_i + t_k \geq w_k + t_i$ и $w_i + t_k \geq w_i \Rightarrow$ готово
 - Случай 1.2: $w_k > w_j + t_k \Rightarrow w_k \geq w_k + t_j \Rightarrow$ невъзможно
- Случай 2: $w_j > w_i + t_j \Rightarrow w_j \geq w_j + t_i \Rightarrow$ невъзможно

**КОНТРОЛНО СЪСТЕЗАНИЕ ЗА РАЗШИРЕН
МОМИЧЕШКИ ОТБОР 27 ЮНИ 2022**
Група G

Оттук получаваме, че проверката на всъщност е еквивалентна на $w_j + t_i \leq w_i + t_j$, но очевидно по-сложното неравенство с максове е по-интуитивно.

Това ни е достатъчно, за да стигнем до заключението, че сортиране спрямо $\leq$ е добре дефинирано и оптимално.

Сложност – $O(N \times \log(N))$.

За подобни задачи за сортиране погледнете:

https://infos.infosbg.com/files/Schools/2025/Vraca/common/lectures/Gatev_greedy_problems.pptx

Четвърта подзадача. Единствено времената на обслужване имат значение. Сортираме клиентите по тях и взимаме първите K .

Сложност – $O(N \times \log(N))$.

Пета подзадача. Тази подзадача е за втората част от решението (сортирането е спрямо входния ред). Малките ограничения и самата постановка навеждат към някакъв вид динамично програмиране. Нека сме фиксирали някакъв префикс от клиентите, които ще обслужим. Наредбата измежду тях не ни интересува, единствено колко на брой са и колко е сумарното им време за обслужване: $t_1 + t_2 + \dots$

Нека $dp[i][k][sumT]$ е минималното време за чакане на k клиента, които трябва да изберем измежду $i, i + 1, \dots, N$, като сумарното време за обслужване на клиентите преди тях е $sumT$. Тогава:

$$dp[i][k][sumT] = \min(dp[i + 1][k][sumT], \max(sumT + w_i, dp[i + 1][k - 1][sumT + t_i]))$$

Съответно това са случаите да не обслужим клиент i и да не го.

Сложност – $O(N \times K \times K \times 250)$.

Шеста подзадача. Комбинираме идеите от трета и пета подзадача.

Сложност – $O(N \times K \times K \times 250)$.

Автор: Александър Гатев