

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА БАГЕРИ

Учатъците, които се копят от един багер, ще разгледаме като отсечки. Например отсечката [3,12] означава всички дупки от номер 3 до номер 12 включително. Тъй като всички багери имат едно и също време за копаене на една дупка, то може да се счита, че ако два багера копаят съседни дупки, няма значение дали се движат един срещу друг или обратното.

Ако багерите могат да изкопаят всички дупки за T минути, те ще могат да я изкопаят и за $T+1$ минути, както и ако не могат за T минути, няма да могат и за $T-1$ минути.

Последното ни навежда на идеята за двоично търсене. За началните стойности L и R на двоичното търсене може да се вземат $(a+b)$ и $(2 \cdot a+b)N$.

Остава да се реши задачата: може ли багеристите да изкопаят всички дупки за T минути за някакво T ? Сортираме багерите във възходящ ред относно позицията им. Ако първият багер копае отсечката $[1, r_1]$, то се получава:

$$l_1 = 1, l_2 = r_1 + 1, l_3 = r_2 + 1, \dots, r_m = n.$$

Нека багерът с номер i копае отсечката $[l_i, r_i]$. Ако той е извън нея ($r_i < l_i$ или $r_i < p_i$), то багерът трябва да иде до най-близката граница на тази отсечка и да започне от нея да копае. Ако той е вътре в отсечката – копае до най-близката граница и после (евентуално) се връща до другата, копаейки останалото. Тогава времето, необходимо на багера, се определя по формулата:

$$t_i = (\min(|p_i - l_i|, |p_i - r_i|) + (r_i - l_i)) \cdot a + (r_i - l_i + 1) \cdot b$$

Нека ни е известно l_i . Как се определя r_i ? Ако багерът за време T може да изкопае отсечката $[l_i; r_i]$, то той може да изкопае и отсечката $[l_i; r_i - 1]$. Т.е. ще добавяме дупки за копаене на багера докато времето му не стане T . Определяйки границите на първия багер автоматично ще знаем лявата граница на втория ($l_2 = r_1 + 1$). Аналогично изчисляваме на втория дясната граница и т.н.

Сортирането на багерите е за $O(M \log M)$, а за определяне на дясната граница за всеки багер може да се използва отново двоично търсене.

Подобрение на алгоритъма:

Имайки предвид, че знаем за i -я багер началната му позиция p_i и лявата граница l_i , има два случая:

- 1) $l_i > p_i$. Багерът отива в лявата граница и копае, движейки се надясно (рис. 1).

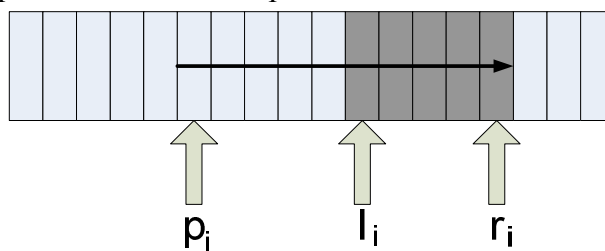


Рис. 1.

- 2) $l_i \leq p_i$. Сега има три варианта:

- 2.1. Багерът тръгва наляво към лявата граница и от едно положение започва да копае (рис. 2).

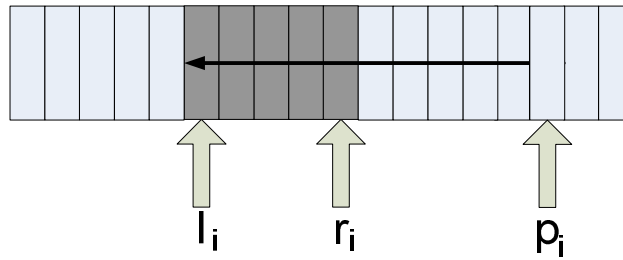


Рис. 2.

2.2. Багерът копае наляво и връщайки се, копае останалото (рис. 3).

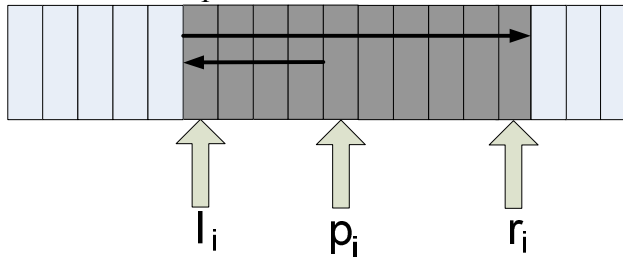


Рис. 3.

2.3. Багерът копае надясно и връщайки се копае останалото (рис. 4).

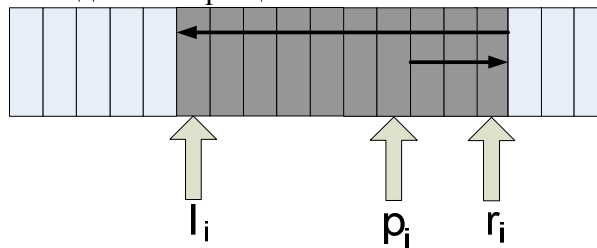


Рис. 4.

Времето за определяне на дясната граница е $O(1)$. Времето за работа на алгоритъма става $O(M \log M + M \log((2a+b)N))$.

Частични решения:

При неголеми N и M може да се използва динамично оптимиране. Нека $F[i, j]$ е минималното време, което е необходимо за първите i багера да изкопае първите j дупки. Тогава формулата е: $F[i, j] = \min(\max(F[i-1, k], \text{Time}(i, k + 1, j)))$ по всички k от 0 до j . Тук с $\text{Time}(i, l_i, r_i)$ е означено времето, необходимо за i -я багер за изкопаване на отсечката $[l_i; r_i]$. Това решение работи за време, пропорционално на $O(N^2M)$ и може да вземе около 40 точки.

Автор: Павел Петров