Zadanie 5 lista 6

Jakub Kuciński, prowadzący Szymon Dudycz

26 czerwca 2020

Spis treści

1 Rozwiązanie

Chcemy znaleźć k-ty największy element. Pivot jest wybierany losowo spośród elementów tablicy. Chcemy oszacować oczekiwany czas działania algorytmu. Czas działania algorytmu jest asymptotycznie równy liczbie porównań elementów tablicy. Niech $y_1, y_2, \ldots, y_n$ będą elementami tablicy w kolejności rosnącej. Zauważmy, że jedynym momentem kiedy elementy są porównywane jest moment, gdy jeden z nich jest pivotem. Po stworzeniu zbiorów elementów większych i mniejszych od pivota możemy sprawdzić, czy pivot jest k-ty co do wielkości, dzięki czemu nie będzie brał udział w dalszym wyszukiwaniu k-tego elementu, więc też nigdy więcej nie zostanie porównany. Wiemy zatem, że każde dwa elementy zostaną porównane co najwyżej raz. Wprowadźmy zmienne losowe

$$X_{i,j} = \begin{cases} 1, & \text{jeśli } y_i \text{ oraz } y_j \text{ zostają porównane przez algorytm} \\ 0, & \text{w przeciwnym przypadku} \end{cases}$$

Zatem oczekiwany czas działania algorytmu to oczekiwana wartość sumy zmiennych losowych $X_{i,j}$.

$$E\left[\sum_{i,j} X_{i,j}\right] = \sum_{i,j} E\left[X_{i,j}\right]$$

Ustalmy k. Rozważmy przypadki:

1. $i < j \leqslant k$
   Jeśli pivotem zostanie wybrany któryś z elementów $y_{k+1}, \ldots, y_n$ to algorytm nie odrzuci ani y_i ani y_j. Czyli wybór takiego pivota nie wpływa na szansę porównania elementów y_i i y_j. Podobnie wybór pivota spośród elementów $y_1, \ldots, y_{i-1}$ nie wpływa na tą szansę. Jeśli jednak wybrany na pivota zostanie któryś z elementów $y_i, \ldots, y_k$ to któryś (być może oba) z elementów y_i i y_j zostanie odrzucony, więc nie będzie mógł zostać już porównany z drugim. Czyli jedyną szansą na porównanie y_i i y_j jest wybór jednego z nich na pivota spośród elementów $y_i, \ldots, y_k$.
   Możemy policzyć sumę wartości oczekiwanych elementów spełniających powyższy warunek.
   $$\sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{2}{k-i+1} = \sum_{i=1}^{k-1} (k-i) \frac{2}{k-i+1} = \sum_{i=1}^{k-1} i \frac{2}{i+1} \leqslant \sum_{i=1}^{k} 2 = 2k = \mathcal{O}(n)$$

2. $k \leqslant i < j$
   Przypadek analogiczny do poprzedniego. Możemy zauważyć, że pytanie o k-ty najmniejszy jest równoważny pytaniu o n-k największy. Wtedy dostaniemy ograniczenie z góry zamiast $2k$ jak w poprzednim przypadku to $2(n-k)$ co jest rzędu $\mathcal{O}(n)$.

3. $i < k < j$ W tym przypadku wybór pivota spośród $y_1, \ldots, y_{i-1}$ lub $y_{j+1}, \ldots, y_n$ nie wpływa na szansę porównania y_i i y_j. Jeśli jednak wybrany na pivota zostanie któryś z elementów $y_i, \ldots, y_j$ to któryś (być może oba) z elementów y_i i y_j zostanie odrzucony, więc nie będzie mógł zostać już porównany z drugim. Czyli jedyną szansą na porównanie y_i i y_j jest wybór jednego z nich na pivota spośród elementów $y_i, \ldots, y_j$.

$$\sum_{i=1}^{k-1} \sum_{j=k+1}^{n} \frac{2}{j-i+1} = \sum_{i=1}^{k-1} (k-i) \frac{2}{k-i+1} = \sum_{i=1}^{k-1} \sum_{j=k-i+1}^{n-i} \frac{2}{j+1} = 2 \sum_{i=1}^{k-1} (H_{n-i+1} - H_{k-i+1}) \leqslant$$
$$\leqslant 2 \sum_{i=1}^{k-1} (\ln(n-i+1) + 1 - \ln(k-i+2)) \leqslant 2 \sum_{i=1}^{k-1} (\ln(n-i+1) + 1 - \ln(k-i)) = 2 \sum_{i=1}^{k-1} (\ln(\frac{n-i+1}{k-i}) + 1) =$$
$$= 2k + 2\ln(\frac{n!}{(k-1)!(n-k+1)!}) = 2k + 2\ln(\binom{n}{k-1}) \leqslant 2k + 2\ln(2^n) = 2k + 2n\ln(2) = \mathcal{O}(n)$$

Suma wszystkich wartości oczekiwanych jest sumą omówionych powyżej przypadków, a skoro wszystkie są rzędu $\mathcal{O}(n)$ to ich suma też jest rzędu $\mathcal{O}(n)$. Zatem oczekiwana złożoność naszego algorytmu wynosi $\mathcal{O}(n)$.