Algoritmi probabiliști. Analiza Quicksort

Problema 1 Secretory problem

Avem n candidați la un job pe care îi intervievăm pe rând.

• Strategia 1

Dacă c_i (candidatul i) este mai bun decât toți candidații anteriori atunci îl/o angajăm și plătim un cost X (generat de concedierea celui anterior)

Câți candidați vom angaja?

În cazul cel mai defavorabil vom angaja n candidați → sortați crescător

• Strategia 2

Permutăm aleator candidații și aplicăm algoritmul anterior.

Care este numărul mediu de candidați angajați?

Vom considera variabile aleatoare cu două valori: 0 și 1

Ex: dat cu banul

$$X = \begin{cases} 1 & \text{dacă pică „pajură"} \\ 0 & \text{altfel} \end{cases}$$

$$P[x=1] = \frac{1}{2} = P[x=0]$$

Media unei variabile aleatoare: $\mathbb{E}[X] = 1 \cdot P[x=1] + \cancel{0 \cdot P[x=0]} = P[x=1]$

expectation

Ex: Dăm de n ori cu banul. Probabilitatea ca moneda să pice „pajură" este p, altfel $1-p$.

Care este numărul mediu în care avem „pajură"

$$X_i = \begin{cases} 1 & \text{dacă la a } i\text{-a aruncare avem „pajură"} \\ 0 & \text{altfel} \end{cases}$$

$X = X_1 + X_2 + \ldots + X_n$

$\mathbb{E}[X] = \mathbb{E}[X_1 + X_2 + \ldots + X_n] = \mathbb{E}[X_1] + \mathbb{E}[X_2] + \ldots + \mathbb{E}[X_n] =$

$= n \cdot \mathbb{E}[X_i] = n(1 \cdot P[X_i = 1] + \cancel{0 \cdot P[X_i = 0]}) = n \cdot P$

$$X_i = \begin{cases} 1 & \text{dacă angajăm candidatul } i \\ 0 & \text{altfel} \end{cases}$$

Vrem să calculăm $X = X_1 + X_2 + \ldots + X_n$

$$\mathbb{E}[X] = \mathbb{E}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathbb{E}[X_i] = \sum_{i=1}^{n} P[X_i = 1] = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n} = O(\log n)$$

Experiment - Birthday Paradox - persoane născute în aceeași zi

$$X_{ij} = \begin{cases} 1 & \text{dacă studentul } i \text{ și studentul } j \text{ sunt născuți în aceeași zi} \\ 0 & \text{altfel} \end{cases}$$

$$X = \sum_{i=1}^{n} \sum_{j=i+1}^{n} X_{ij}$$

$$\mathbb{E}[X] = \mathbb{E}\left[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}\right] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbb{E}[X_{ij}] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{365} = \frac{n(n+1)}{2 \cdot 365} = \frac{50 \cdot 51}{2 \cdot 365}$$

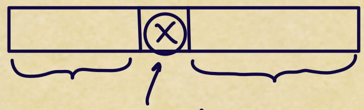
$$P[X_{ij} = 1] = \sum_{z=1}^{365} P[\text{studentul } i \text{ să fie născut în ziua } z \;\&\; j \text{ să fie născut în ziua } z] =$$

$$= 365 \cdot P(\text{studentul } i \text{ să fie născut în ziua } z) \cdot P(j \text{ să fie născut în ziua } z) =$$

$$= 365 \cdot \frac{1}{365} \cdot \frac{1}{365} = \frac{1}{365}$$

QuickSort: Cap 5 & 7 Cormen

Ideea de bază:



pe poziția finală în vectorul sortat

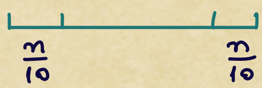
1. Alegem un pivot
2. Apelăm funcția de partiție care așează pivotul pe poziția finală în vectorul sortat $\quad O(n)$
3. Apelăm recursiv quicksort pe șirul din stânga, respectiv din dreapta pivotului

Pivot determinist $T(n) = T(n-1) + O(n) = O(n^2)$

Cazul defavorabil

În cazul favorabil: $T(n) = 2T\left(\frac{n}{2}\right) + O(n) = O(n \log n)$

Ex:



Dacă am avea $T(n) = T\left(\frac{n}{10}\right) + T\left(\frac{9n}{10}\right) + O(n) = O(n \log n)$

Similar $T(n) = T\left(\frac{n}{1000}\right) + T\left(\frac{999n}{1000}\right) + O(n) = O(n \log n)$

Notăm elementele $z_1, z_2, \dots, z_n$ unde

$$z_1 \le z_2 \le \dots \le z_n$$

$$x_{ij} = \begin{cases} 1 & \text{dacă } z_i \text{ este comparat cu } z_j \\ 0 & \text{altfel} \end{cases}$$

X = numărul de comparații efectuat de algoritm $= \sum_{i=1}^{n-1} \sum_{i=j+1}^{n} x_{ij}$

$$\mathbb{E}[X] = \mathbb{E}\left[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} x_{ij}\right] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbb{E}[x_{ij}] =$$

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} P[x_{ij} = 1]$$

$P[x_{ij} = 1] = P[z_i$ sau z_j să fie aleși ca pivot din intervalul $z_i, z_{i+1}, \dots, z_j] = \frac{2}{j-i+1}$

$z_i, z_{i+1}, \dots, z_j$

$$E[x] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1} = \sum_{i=1}^{n} \sum_{k=1}^{n-i} \frac{2}{k+1} < \sum_{i=1}^{n} \sum_{k=1}^{n} \frac{2}{k+1}$$

$$k = j - i$$

$$\approx O(\log n)$$

$$= O(n \log n)$$