Randomized Algorithms & Probabilistic Analysis

(Hire Assistant Problem)

<u>HireAssistant(n)</u>

best ← 0
for i = 1 (1) n
    interview(i) — c
    if (S_i > best)
        hire(i) — c^*
        best ← S_i

$$\text{Cost} = O(n \cdot c + n \cdot c^*)$$

Randomize the order in which the candidates are interviewed

Indicator Variables

$$I_A = \begin{cases} 1, & \text{if } A \text{ occurs} \\ 0, & \text{o/w.} \end{cases} \qquad \mathrm{Exp}(I_A) = \Pr[A \text{ occurs}]$$

Coin toss. Expected no. heads in one toss.

$I_H = 1$, if outcome is head
$= 0$, o/w

$$\mathrm{Exp}(I_H) = 1 \cdot \Pr[H] + 0 \cdot \Pr[T]$$
$$= \Pr[H] = \frac{1}{2}$$

$$\begin{cases} X_i = 1, & i^{th} \text{ candidate is selected} \\ \quad = 0, & o/w \end{cases}$$

$$\underbrace{1000 \cdot c^*}$$

$$\underbrace{10 \cdot c^*}$$

$$\#\ \text{hire}_1(i) \text{ executed} = X = X_1 + X_2 + \cdots + X_n$$

$$\begin{aligned} Exp(X) &= Exp\left(\sum_{i=1}^{n} X_i\right) \\ &= \sum_{i=1}^{n} Exp(X_i) \\ &= \sum_{i=1}^{n} \frac{1}{i} = \log n \end{aligned}$$

$$\left\{ \begin{aligned} Exp(X_i) &= Pr[i^{th} \text{ candidate is hired}] \\ &= \frac{1}{i} \end{aligned} \right.$$

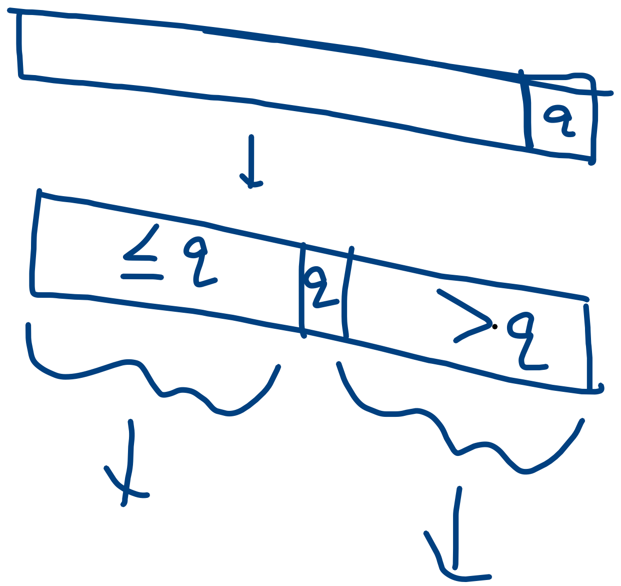
Expected Running Time of Quicksort

Partition(A, p, r)

$x = A[r]$
$i = p - 1$
for $j = p\ (1)\ r-1$
  if $A[j] \le x$ (*)
    $i += 1$
    Swap(A[i], A[j])
Swap(A[i+1], A[r])
return $i+1$

\$-Partition(A, p, r)

i = Random(p, r)
Swap(A[i], A[r])
Partition(A, p, r)



\$-Quicksort(A, p, r)

if $p < r$
  q = \$-Partition(A, p, r)
  \$-Quicksort(A, p, q-1)
  \$-Quicksort(A, q+1, r)

Claim: Suppose, you are applying the ~~S~~- Quick Sort algorithm on n-element array (distinct) & line $(*)$ gets executed exactly "X" times, then the running time is $O(n+X)$.

Prove: H/W.

Goal: Estimate X.

$A \to z_1, \ldots, z_n$ s.t. $z_1 < z_2 < \cdots < z_n\}$

$Z_{ij} = \{z_i, z_{i+1}, \ldots, z_j\}$

$$X_{ij} = \begin{cases} 1, & z_i \text{ is compared with } z_j \\ 0, & \text{o/ws} \end{cases}$$

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

$$E(X) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E(X_{ij})$$

$E(X_{ij})$

$= \Pr[z_i \text{ is compared to } z_j]$

Goal: Estimate X.

$A \to z_1, \ldots, z_n$ s.t. $z_1 < z_2 < \cdots < z_n$ }

$Z_{ij} = \{z_i, z_{i+1}, \ldots, z_j\}$

$E(X_{ij})$

$= \Pr[z_i$ is compared to $z_j]$

$$X_{ij} = \begin{cases} 1, & z_i \text{ is compared with } z_j \\ 0, & \text{o/ws} \end{cases}$$

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

$$E(X) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E(X_{ij})$$

Goal: Estimate X.

$A \rightarrow z_1, \ldots, z_n$ s.t. $z_1 < z_2 < \cdots < z_n \}$

$Z_{ij} = \{ z_i, z_{i+1}, \ldots, z_j \}$

$$X_{ij} = \begin{cases} 1, & z_i \text{ is compared with } z_j \\ 0, & \text{o/ws} \end{cases}$$

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

$$E(X) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E(X_{ij})$$

$E(X_{ij})$

$= \Pr[z_i \text{ is compared to } z_j]$

$= \Pr[z_i \text{ or } z_j \text{ is selected as the pivot in } Z_{ij}]$

Goal: Estimate X.

$A \to z_1, \ldots, z_n$ s.t. $z_1 < z_2 < \cdots < z_n$

$Z_{ij} = \{z_i, z_{i+1}, \ldots, z_j\}$

$$X_{ij} = \begin{cases} 1, & z_i \text{ is compared with } z_j \\ 0, & \text{o/ws} \end{cases}$$

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

$E(X_{ij})$

$= \Pr[z_i$ is compared to $z_j]$

$= \Pr[z_i$ or z_j is selected as the pivot in $Z_{ij}]$

$= \dfrac{2}{j-i+1}$

$$E(X) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E(X_{ij})$$

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1} \le \sum_{i=1}^{n-1} \boxed{\sum_{k=1}^{n} \frac{2}{k}}$$

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

When z_i is compared to z_j?

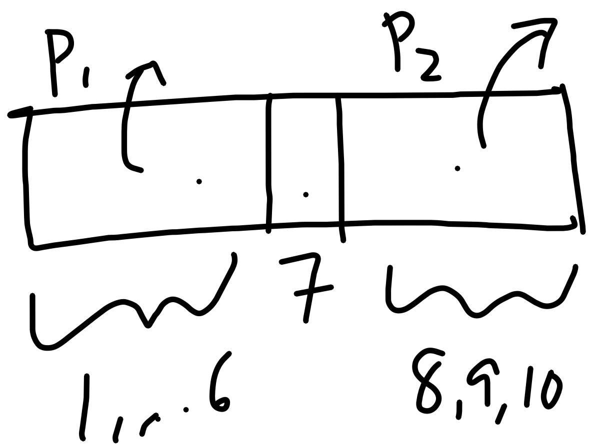
- At most 1 comparison
- One of z_i & z_j must be a pivot element in one of the executions.
- In $Z_{ij} = \{z_i, \ldots, z_j\}$ iff either z_i or z_j should be the pivot element.

$z_i < z_k < z_j$

Suppose z_k is a pivot element.
Can z_i & z_j be compared?

$\{1, 2, \ldots, 10\}$

Suppose 7 is pivot

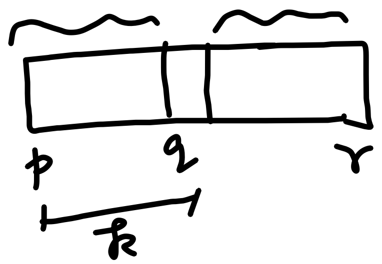


Two elements, one from P1 & one from P_2 is never compared.

Selection in Linear Time

\$-Select(A, p, r, i)

```
if p == r
    return A[p]
q = $-Partition(A, p, r)
k = q - p + 1
if i == k
    return A[q]
else if i < k
    return $-Select(A, p, q-1, i)    → T(k-1)
else
    return $-Select(A, q+1, r, i-k)  → T(n-k)
```



$$X_k = \begin{cases} 1, & \text{if the subarray } A[p..q] \text{ has exactly } k \text{ elements} \\ 0, & \text{o/w} \end{cases}$$

$$T(n) = \sum_{k=1}^{n} X_k \max\{T(k-1), T(n-k)\} + O(n)$$

{Array of n elements}

$$E(T(n)) = \sum_{k=1}^{n} E[X_k] \cdot E[T(\max(k-1, n-k))]$$

$$= \sum_{k=1}^{n} \frac{1}{n} \cdot E[T(\max(k-1, n-k))] + O(n)$$

$$= \frac{1}{n} \sum_{k=1}^{n} E\left(T\left(\max(k-1, n-k)\right)\right)$$

$T(n)$

$$\le \frac{2}{n} \sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} E\left(T(k)\right) + O(n)$$

Assume

$$\boxed{T(n) = c \cdot n}$$

Solve the recurrence using Substitution.

$$T(n) \le \frac{2}{n} \sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} c \cdot k + \alpha n$$

$$cn \le$$

$k=1 \Rightarrow n-1$

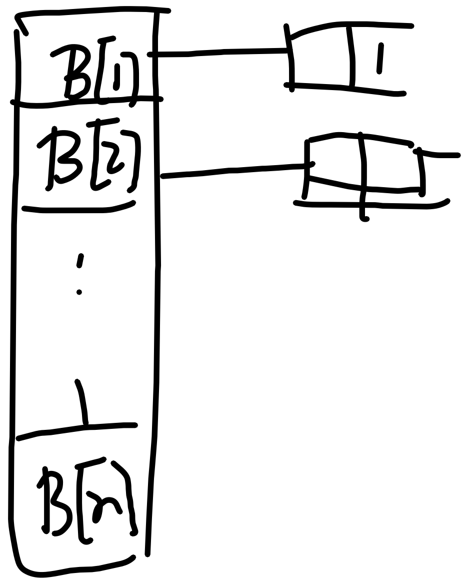
$k=2 \Rightarrow n-2$

$\vdots$

$k=n \Rightarrow n-1$

$k=n-1 \Rightarrow n-2$

Running Time for Bucket Sort



$N_i \rightarrow$ R/V denoting # elements placed in Bucket $B[i]$.

$$T(n) = \Theta(n) + \sum_{i=1}^{n} O(N_i^2)$$

$$E(T(n)) = O(n) + E\left(\sum_{i=1}^{n} O(N_i^2)\right)$$

$$X_{ij} = 1, \text{ if } A[j] \text{ falls in bucket } B[i]$$
$$= 0, \text{ o/w}$$

$$N_i = X_{i,1} + X_{i,2} + \cdots + X_{i,n}$$