EECS 311: Data Structure and Data Management **Lecture 28**
Quicksort

Quicksort

"the fastest, most practical sorting routine"

Algorithm: Quicksort
Input: unsorted set S of n elements.

0. if $|S| \leq 1$, output S.
1. pick "pivot" element $x \in S$.
2. partition S into $S' = \{y \; : \; y < x\}$ and $S'' = \{y \; : \; y \geq x\}$
3. output: Quicksort(S'), x, Quicksort(S'').

Claim: Quicksort returns the elements in sorted order.

Proof: by induction on $|S|$.

Claim: if quicksort always picks smallest element as pivot, runtime is n^2

Proof:

- let $T(n)$ be runtime on size n input.
- Base Case: $T(1) = 1$.
- Recursive Case: $T(n) = T(n-1) + n$
- Solve for $T(n)$:

$$\begin{aligned}
T(n) &= T(n-1) + n \\
&= T(n-2) + (n-1) + n \\
&= T(n-3) + (n-2) + (n-1) + n \\
&\vdots \\
&= T(1) + 2 + 3 + \cdots + (n-1) + n \\
&= \sum_{i=1}^{n} i \\
&= \binom{n}{2} \approx n^2.
\end{aligned}$$

□

Claim: if quicksort always picks median as pivot, runtime is $\approx n^2$.

Proof:

- let $T(n)$ be runtime on size n input.
- Base Case: $T(1) = 1$.
- Recursive Case: $T(n) = 2T(n/2) + n$
- Solve for $T(n)$

$$\begin{aligned}
T(n) &= T(n/2) + n \\
&= 2(2T(n/4) + n/2) + n \\
&= 4T(n/4) + n + n \\
&= 4(2T(n/8) + n/4) + n + n \\
&= 8T(n/8) + n + n + n \\
&\vdots \\
&= nT(1) + n + n + \cdots + n \\
&= n \log n
\end{aligned}$$

□

Idea: pick pivot randomly!!

Claim: if quicksort picks pivot randomly, expected runtime is $n \log n$.

Proof:

- runtime = number of comparisons.
- $X_{ij} = \begin{cases} 1 & \text{if } i\text{th and } j\text{th biggest compared} \\ 0 & \text{otherwise} \end{cases}$
- number of comparisions = X = $\sum_{i<j} X_{ij}$
- $\mathbf{E}[\text{runtime}] = \mathbf{E}[X]$:

$$\mathbf{E}[X] = \mathbf{E}\left[\sum_{i<j} X_{ij}\right] = \sum_{i<j} \mathbf{E}[X_{ij}]$$

- Subclaim: $\mathbf{E}[X_{ij}] = \frac{2}{j-i+1}$
  - i and j only compared when one is pivot.
  - if k with $i < k < j$ picked as pivot before i or j, never compare i with j.
  - otherwise, compare i with j when i or j picked as pivot.
  - $\mathbf{Pr}[i \text{ or } j \text{ picked before } k] = \frac{2}{j-i+1}$.
  - $\mathbf{E}[X_{ij}] = 1 \times \mathbf{Pr}[X_{ij} = 1] = \mathbf{Pr}[i \text{ or } j \text{ picked before } k]$.
- Calculate $\mathbf{E}[X]$:

$$\mathbf{E}[X] = \sum_{i<j} \mathbf{E}[X_{ij}] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1}$$

change variable of sum, $d = j - i + 1$, so

$$\mathbf{E}[X] = \sum_{i=1}^{n-1} \sum_{d=2}^{n-i+1} \frac{2}{d+1}$$

But $\sum_{k=1}^{n} 1/k \approx \log n$

$$\mathbf{E}[X] \approx 2 \sum_{i=1}^{n-1} \log(n - i + 1) \approx 2 \sum_{j=1}^{n} \log j$$

so certainly

$$n/2 \log n/2 \leq \mathbf{E}[X] \leq n \log n$$

so

$$\mathbf{E}[X] \approx n \log n.$$

□