Primality testing: Given an integer $N$ is $N$ prime?

For $i = 1, 2, \sqrt{N}$ if $i$ divides $N$

$O(\sqrt{N})$ Input size: $\log n$ bits $\frac{n}{2} = n$ density of primes: among $n$ integers $n \sim \frac{n}{\ln n}$ are prime

Pick a large random integer $k$ test if $k \times n$ prime

Efficient algorithm for primality test:
Rabin–Miller
Solovay–Strassen
(Monte Carlo) Randomized: $n^2$
AKS algorithm deterministic: $n^{12}$
Selection Problem
(totally ordered)

Given \( n \) elements from a set \( S \) and an integer \( 1 \leq k \leq n \), find an element \( x \in S \) such that \( \text{rank}(x, S) = k \)

Objective: Design a linear-time algorithm for selection

Easy soln: do sort

Pick some element \( r \in S \) and partition \( S \) using \( r \)

Prime and search

\[ S \]

\[ r \]

\[ S_1 \]

\[ S_2 \]

\[ S_3 \]
Analyzing: If the subset containing the rank $k$ element decreases by some constant factor, say $d < 1$ running time: $\leq d^i n = O(n)$

Approximate median: If the rank of $r$ is between $\left[ \frac{1}{4} n, \frac{3}{4} n \right]$

$r$: random element $

r$ is an approximate median with $\Pr[|b| \geq \frac{1}{2}]$

Find rank of $r$

If $\text{rank } r \in \left[ \frac{1}{4} n, \frac{3}{4} n \right]$

then proceed with the correct partition

else
Expected # of iterations = 2

In each iteration we spend $cn$ time.

First recursive call: $2cn X_1$
semi.

$$i = 2c \cdot \left(\frac{3}{4}\right)^i n X_m$$

$$m = \log_{\frac{3}{4}} n$$

$O(cn)$

$X$: random variable representing the overall running time

$$E[X] = E[cn \cdot X_1 + c \cdot \left(\frac{3}{4}\right)n \cdot X_2 + \cdots (\cdot )X_m]$$

using linearity of expectation

$$E[x+y] = E[x] + E[y]$$

for any r.v. $X, Y$ (not necessarily independent).

$E[X] = O(cn)$
Not repeating sampling

Median of medians for approximate median

\[ \text{Choose the median of medians, and that is } R \]
\[ T(n) \leq T\left(\frac{n}{5}\right) + T\left(\frac{7n}{10}\right) + O(n) \]

Solve by guessing \( T = cn \) for some constant \( c \)

\[ \sim c \]

Analyze the space complexity of the algorithm.