Randomized Quicksort

A tiny bit more complicated than Quicksort

we don't know which side the element will lie on

Instead of having

\[ X_n = X_{i-1} + X_{n-i} + n - 1 \]

the recursion is now

\[ \max(X_{i-1}, X_{n-i}) \]

Divide the problem into levels, \( L_1, L_2, \ldots \)

where \( h_j \rightarrow \) size of the problem at level \( j \)

Claim

\[ E[h_i] = \left( \frac{3}{4} \right)^i n \quad \text{for } i = 0, 1, 2, \ldots \]

Proof: at level 1,

\[ h_1 = \max(n-i, i-1) \text{ when } x_i \text{ is pivot.} \]

\[ \Rightarrow E[h_i] = E[\max(n-i, i-1)] \]

\[ = \frac{1}{n} \sum_{i=1}^{n} \max(n-i, i-1) \]

\[ \leq \frac{1}{n} \frac{3}{4} n^2 = \frac{3}{4} n \]
for proof of sum -> area under the wax curve

\[ E[h_{k-1}] \leq \left( \frac{3}{4} \right)^{k-1} \cdot n \]

Suppose by induction

\[ E[h_k] = \sum_{\omega} E[h_k | h_{k-1} = \omega], \Pr(h_{k-1} = \omega) \]

\[ = \sum_{\omega} \frac{3}{4} \cdot \omega \cdot \Pr(h_{k-1} = \omega) \]

\[ = \frac{3}{4} \sum_{\omega} \Pr(h_{k-1} = \omega) \cdot \omega \]

\[ = \frac{3}{4} \cdot \Pr(h_{k-1}) \]

\[ = \frac{3}{4} \cdot E[h_{k-1}] \]

\[ \leq \left( \frac{3}{4} \right)^k \cdot n \]
This is called the Markov property

"History-less"

Size of k-th layer does not depend on anything except k=1 — history does not matter.

\[ \Rightarrow \text{Expected running time} \]

\[ = \sum_{j=0}^{\infty} E[h_j] \leq \sum_{j=0}^{\infty} \left( \frac{3}{4} \right)^j \cdot n \]

\[ = n \sum_{j=0}^{\infty} \left( \frac{3}{4} \right)^j = 4n \]

\[ \Rightarrow \text{Quickselect} = O(n) \text{ in expected time.} \]

Analysis of depth.

D \rightarrow \text{depth of quickselect recursion}

\[ E[D] = \sum \Pr(D \geq j) \]

D \geq j \Leftrightarrow h_j > 1

By Markov's inequality \( \Pr(h_j \geq 1) \leq \frac{E[h_j]}{\left( \frac{3}{4} \right)^j \cdot n} \)
let \( d_0 = \lceil \log_{3/4} n \rceil + 1 \)

\[
= \sum_{j=1}^{\infty} P_X(D \geq j) \\
= \sum_{j=1}^{d_0-1} P_X(k_j \geq 1) + \sum_{j=d_0}^{\infty} P_X(k_j \geq 1) \\
\leq d_0 - 1 + \sum_{j=d_0}^{d_0+1} \left( \frac{3}{4} \right)^j \cdot n \\
= d_0 - 1 + \left( \frac{3}{4} \right)^{d_0+1} \cdot n \sum_{j=d_0}^{\infty} \left( \frac{3}{4} \right)^j \\
\leq d_0 - 1 + 4 = O \left( \log_{3/4} n \right)
\]

Using Markov's Inequality

\rightarrow \text{depth is almost } O \left( \log_{3/4} n \right)
Types of randomized algorithms.

1) Las Vegas algorithms
* Always give the correct answer
* Running time is randomized.
  e.g. quicksort / quickselect.

2) Monte Carlo
* Running time is fixed
* May or may not give the correct answer.
  e.g.
* Given a list of numbers $a_1, \ldots, a_n$
  find the largest $g \geq 1$ that divides at least half the
  numbers in $a_1, \ldots, a_n$ — $\gcd$.

Additional constraint — do it in constant memory
  (almost — constant)

Pre-requisites.
Fact → Given a number $a$ — you can factorize it only
  very slowly.

Cryptography relies on this.
For the purposes of this question
- assume factorization can be done in \(O(1)\)

Algorithm
- choose a random \(a_i\)
- for all of its factors \(f_1, \ldots, f_k\)
  - check if any of them divide more than half
- return the largest

Let us look at the probability of getting the correct answer.

Suppose that the GCD divides \(a_1, \ldots, a_n\) whole.
then \(a_i \in \{a_1, \ldots, a_n/2\} \) with probability \(1/2\)

\[ \Rightarrow \text{if } a_i \in \{a_1, \ldots, a_n/2\} \text{ we will have the GCD as one of the factors.} \]

\[ \Rightarrow \text{Probability of success } \geq \frac{1}{2} \]

Monte-Carlo repeated starts.
Run the above \(k\) times
\[ \Rightarrow \text{Probability of failure } \leq \frac{1}{2^k} \]
\[ \Rightarrow \text{correct answer with } 1 - \frac{1}{2^k} \]
time complexity analysis.

1) max # of factors = \log_2 a^i \quad (\text{if } a^i = 2^k)

=) total runtime = \sum \log a^i

over K starts

\leq O (Kn \log \text{max } a^i) \\& \text{success } = 1 - \frac{1}{2^K}

\underline{Minimum-Cut \text{ in a graph}}

given a graph \( G = (V, E) \)
a cut = \( A, B \subseteq V \) \text{ s.t. } A \cap B = \emptyset, A \cup B = V.

i.e. a partition of the vertices.

E.g.

```
A
\begin{tikzpicture}
\node[draw] (1) at (0,0) {1};
\node[draw] (2) at (0,-1) {2};
\node[draw] (3) at (0,-2) {3};
\node[draw] (4) at (1,0) {4};
\node[draw] (5) at (1,-1) {5};
\node[draw] (6) at (1,-2) {6};
\node[draw] (7) at (0,-3) {7};
\draw (1) -- (2) -- (3) -- (4) -- (5) -- (6) -- (7);
\end{tikzpicture}
```

```
B
```

the size of the cut = \# edges betw \( A \) \& \( B \)

= 4 \quad \text{for given graph}
Given a graph find the minimum-cut of $G$.

$\rightarrow$ Deterministic algorithm $\rightarrow V^2E, V^3, \ldots$

Karger Stein - Randomized algo $\rightarrow O(n^2 \log^3 n)$

---

Preliminary Algo - Karger's contraction

Contraction of an edge

Given $G, e \in E$

$G/e$ : graph obtained by contracting $e$ in $G$.

e.g.

$G$:

\begin{align*}
  &A \\
  &\quad \downarrow \\
  &B \\
  &\quad \downarrow \\
  &C \\
  &\quad \downarrow \\
  &D
\end{align*}

$G/AB$:

\begin{align*}
  &AB \\
  &\quad \downarrow \\
  &C \\
  &\quad \downarrow \\
  &D
\end{align*}

$G/AB$:

\begin{align*}
  &AB \\
  &\quad \downarrow \\
  &C \\
  &\quad \downarrow \\
  &D
\end{align*}

-- no longer a simple graph

multi-graph - multiple edges, both nodes.
Intuitive idea
if we only contract edges in A or B, the cut remains the same.

Karger's algorithm
while |V| > 2
    select a random edge e
    |V| ← |V|/e

Claim: Karger's algorithm finds the minimum cut with probability \( \geq \frac{1}{n^2} \)

Proof: Suppose that the size of the minimum cut = K.
The algorithm will succeed if at each point we select an edge not in the cut.
The probability of this happening = \( 1 - \frac{K}{|E|} \)

Now as \( \text{min cut} = K \)
minimum degree of any vertex \( \geq K \)
if not, say \( v \) has \( \leq K-1 \) nbh then
\( A \) \( \cup \) \( B \) is a cut of size \( \leq K-1 \)
\( \Rightarrow \in \)
\[=) \text{ degree of each vertex } \geq K\]

\[=) \# \text{ edges } \geq \frac{nk}{2}\]

\[\Rightarrow |E| \geq \frac{nk}{2}\]

\[\Rightarrow \frac{k}{|E|} \leq \frac{K}{\frac{nk}{2}} = \frac{2}{n}\]

\[\Rightarrow 1 - \frac{k}{|E|} \geq 1 - \frac{2}{n}\]

\[\therefore \text{ at each step of the algorithm we should choose an edge from the min cut}\]

\[\text{Probability of not selecting an edge in min-cut at each stage}\]

\[= \prod_{i=0}^{n-3} \left( 1 - \frac{2}{n-i} \right)\]

\[= \frac{n-3}{n-1} \cdot \frac{n-2}{n-1} \cdot \frac{n-1}{n-1} \cdot \frac{n-2}{n-1} \cdot \ldots \cdot \frac{4}{6} \cdot \frac{3}{5} \cdot \frac{2}{4} \cdot \frac{1}{2}\]

\[= \frac{2}{n(n-1)} = \frac{1}{n(n-1)/2} = \frac{1}{nC_2}\]
by repeating it $K$ times

$\text{success probability} = \left( 1 - \frac{1}{n_{c_2}} \right)^K$

$(1 - \frac{1}{x})^x \to e$

$x \to \infty$

$\Rightarrow K = n_{c_2}$

$\text{success probability} = \left( 1 - \frac{1}{n_{c_2}} \right)^{n_{c_2}} = \frac{1}{e}$