Randomization

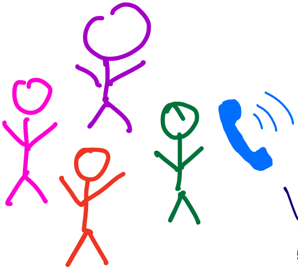
5112

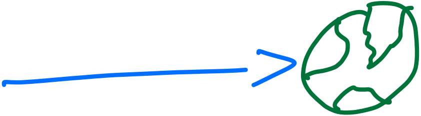
Shared Channel

QuickSelect

QuickSort

Shared Channel





We all want to send a message, but if more than one person sends a message at a time, the messages don't send.

n clients

At each time t, a client can either try to send a message or not.

No feedback: can't tell if ~~your~~ any message sends

Each client has to execute the same algorithm
No communication!

Deterministic algorithms never send a single message.

A Randomized Algorithm

At each timestep send the message with probability p.

Define $A(i,t)$ to be the event that client i sends a message at time t.

$\overline{A(i,t)}$, the complement of $A(i,t)$: the event where $A(i,t)$ didn't happen.

$Pr(A(i,t)) = p \quad \forall\, i$ and t. $\quad Pr(\overline{A(i,t)}) = 1-p$

Probability Spaces and Events

Example roll a six-sided die:

Event : Roll a 5 $= \{5\}$

Roll at least a 3 $= \{3, 4, 5, 6\}$.

Example : Flip two coins.

Outcomes: $\{HH, HT, TH, TT\}$

Blue comes up heads: $BH = \{HH, HT\}$

Red comes up heads: $RH = \{HH, TH\}$

Both are heads:

AND of BH and RH

$BH \cap RH$

$= \{HH\}$

Probability Spaces and Events

Events A and B are independent if

$$\Pr(A \cap B) = \Pr(A)\Pr(B).$$

What's the probability that client i succeeds at time t.

Client i succeeds if 1. client i sends a message
2. no one else does

$S[i,t]$ is the event that client i succeeds at time t.

$$S[i,t] = A[i,t] \cap \left(\bigcap_{j \neq i} \overline{A[j,t]} \right)$$

$$\Pr\{S[i,t]\} = \Pr\{A[i,t]\} \cdot \prod_{j \neq i} \Pr\{\overline{A[j,t]}\}$$

$$= p(1-p)^{n-1}$$

$$\Pr[S[i,t]] = p(1-p)^{n-1}$$

What's the best choice of p. Want to maximize.

$$\frac{d}{dp}\, p(1-p)^{n-1} = 1 \cdot (1-p)^{n-1} - p(n-1)(1-p)^{n-2} = 0$$

Solve for p: $(1-p)^{n-1} = p(n-1)(1-p)^{n-2}$

$$1-p = p(n-1) = pn - p$$

$$pn = 1$$

$$p = \frac{1}{n}$$

Subst $p = \frac{1}{n}$

$$\Pr\{S[i,t]\} = p(1-p)^{n-1} = \frac{1}{n}\underline{\left(1-\frac{1}{n}\right)^{n-1}}$$

Theorem

1. $\left(1-\frac{1}{n}\right)^{n-1}$ converges monotonically from $\frac{1}{2}$ to $\frac{1}{e}$
2. $\left(1-\frac{1}{n}\right)^{n}$ converges monotonically from $\frac{1}{4}$ to $\frac{1}{e}$

This implies $\frac{1}{en} \leq \Pr\{S[i,t]\} \leq \frac{1}{2n}$, i.e. $\Pr\{S[i,t]\} = \Theta\left(\frac{1}{n}\right)$

Define $F[i,t]$ to be the event that client i doesn't succeed in <u>any</u> round from $1, \ldots, t$.

$$F[i,t] = \bigcap_{r=1}^{t} \overline{S[i,r]}$$

$$\Pr[F[i,t]] = \prod_{r=1}^{t} \Pr[\overline{S[i,r]}]$$

$$\leq \left[1 - \frac{1}{en}\right]^{t}$$

$$= \left[1 - \frac{1}{en}\right]^{\lceil en \rceil}$$

$$\leq \left[1 - \frac{1}{en}\right]^{en} \leq \frac{1}{e}$$

Want this to look like $\left(1 - \frac{1}{n}\right)^n$.

Set $t = \lceil en \rceil$

Set $t = \lceil en \rceil \cdot \lceil c \ln n \rceil$

$$\Pr\{F[i,t]\} \le \left(1 - \frac{1}{en}\right)^t = \left(\left(1 - \frac{1}{en}\right)^{\lceil en \rceil}\right)^{c \ln n}$$

$$\le \left(\frac{1}{e}\right)^{c \ln n}$$

$$= e^{-c \ln n} \qquad (e^{\ln n} = n)$$

$$= n^{-c} = \frac{1}{n^c}$$

After $t = \lceil en \rceil \cdot \lceil c \ln n \rceil$ rounds any given client has succeeded w/ prob. $\ge 1 - \frac{1}{n^c}$

Define $F_t = \bigcup_{i=1}^{n} F[i,t]$

Prob. that <u>any</u> client hasn't succeeded in any round $1,\dots,t$.

Theorem (Union Bound)

Given events $A_1, A_2, \dots, A_n$

$$\Pr\left[\bigcup_{i=1}^{n} A_i\right] \le \sum_{i=1}^{n} \Pr\{A_i\}$$

$$\Pr\{F_t\} = \Pr\left[\bigcup_{i=1}^{n} F[i,t]\right] \le \sum_{i=1}^{n} \Pr\{F[i,t]\} \le \frac{n}{n^c} \le \frac{1}{n^{c-1}}$$

if we go $t = \lceil en \rceil \cdot \lceil c \ln n \rceil$ rounds

$t = O(n \log n)$

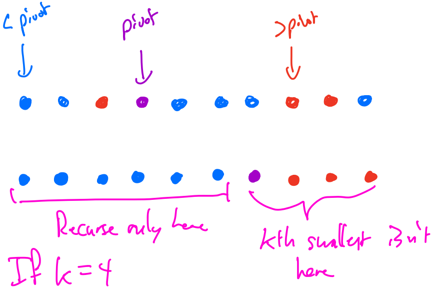
Median (Or Selection)

Input: Array & n numbers.
Output: the k^{th} smallest

Simple solution: Sort then return k^{th} index
($O(n \log n)$ algorithm)

QuickSelect is a randomized $O(n)$ algorithm.

Idea: Choose a random pivot.
Use divide and conquer.
Only recurse on one side



QuickSelect(A, k):
// A has length n
// $1 \le k \le n$

Choose $i \in \{1, \dots, n\}$
Form arrays B and C
for each $j = 1, \dots, n$, $j \ne i$
If $A[j] \le A[i]$
    Append $A[j]$ to B
Else
    Append $A[j]$ to C.
Let n_B and n_C be the lengths

QuickSelect(A, k):
// A has length n
// $1 \leq k \leq n$

Choose $i \in \{1, \dots, n\}$
Form arrays B and C
For each $j = 1, \dots, n$, $j \neq i$
If $A[j] \leq A[i]$
    Append $A[j]$ to B
Else
    Append $A[j]$ to C.
Let n_B and n_C be the lengths of B and C
If $n_B = k-1$
    Output $A[i]$
Else if $n_B \geq k-1$
    Output QuickSelect$(B, k-1)$
Output QuickSelect$(C, k-1-n_B)$