Randomized Algorithm: allow decisions based on coin flips

RP

for some constant $c > 0$

Algorithm A solves X with one-sided error if $X(x) = \text{YES} \rightarrow A(x) = \text{YES}$ with prob $p > c$
$X(x) = \text{NO} \rightarrow A(x) = \text{NO}$ with prob $= 1$

two-sided error $X(x) = \text{YES} \rightarrow A(x) = \text{YES}$ with prob $p > \frac{1}{2}$
$X(x) = \text{NO} \rightarrow A(x) = \text{NO}$ with prob $p > \frac{1}{2}$

BPP

ZPP: decision problems with solutions with polynomial expected running time
(but possibly unbounded worst case)

$$P \subseteq ZPP \subseteq NP$$

don't know if these are proper

(and if some NP-complete X satisfies $X \in ZPP$, then $ZPP = NP$)

Median

FIND-MEDIAN: Given array A of distinct integers, return the median

FIND-MEDIAN-SORT(A)
$n \leftarrow$ len(A)
MERGE-SORT(A)
return $A[\frac{n}{2}]$

$\Theta(n \log n)$ comparisons

lower bound: $2n$ comparisons

divide-and-conquer : $< 5.44n$ comparisons

Dor and Zwick : $< 2.95n$ comparisons

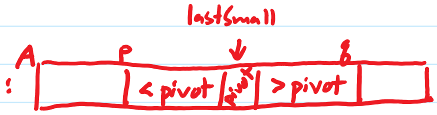
$k=0 \rightarrow$ min
$k=n-1 \rightarrow$ max
$k=\frac{n}{2} \rightarrow$ median

SELECT: Given array A of distinct integers and k, return the element of rank k

```
SELECT(A, p, q, k)
  if p=q return A[p]
  r ← PARTITION(A, p, q)
  if k = r
     return A[p+r]
  else if k<r
     return SELECT(A, p, p+r-1, k)
  else
     return SELECT(A, p+r+1, q, k-r-1)
```

```
PARTITION(A, p, q)

pivot ← A[p]
lastSmall ← p
for i = p+1 to q
    if A[i] < pivot
        lastSmall ← lastSmall + 1
        swap(A, i, lastSmall)
swap(A, p, lastSmall)
return lastSmall - p
```

POST:


n-1 element comparisons

if 50/50 split: $T(n) = T(\frac{n}{2}) + O(n) \longrightarrow T(n)$ is $\Theta(n)$

worst case (sorted): $T(n) = T(n-1) + O(n) \longrightarrow T(n)$ is $\Theta(n^2)$

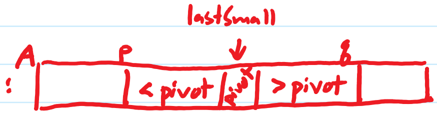
Select

$k = 0 \rightarrow$ min
$k = n-1 \rightarrow$ max
$k = \frac{n}{2} \rightarrow$ median

SELECT: Given array A of distinct integers and k, return the element of rank k

```
RANDOMIZED-SELECT(A, p, q, k)
  if p=q return A[p]
  r ← RANDOMIZED-PARTITION(A, p, q)
  if k = r
     return A[p+r]
  else if k<r
     return SELECT(A, p, p+r-1, k)
  else
     return SELECT(A, p+r+1, q, k-r-1)
```

```
RANDOMIZED-PARTITION(A, p, q)
  swap(A, p, random(p, q))
  pivot ← A[p]
  lastSmall ← p
  for i = p+1 to q
     if A[i] < pivot
        lastSmall ← lastSmall + 1
        swap(A, i, lastSmall)
  swap(A, p, lastSmall)
  return lastSmall - p
```

POST :


n-1 element comparisons

if 50/50 split: $T(n) = T(\frac{n}{2}) + O(n) \longrightarrow T(n)$ is $\Theta(n)$

worst case (sorted): $T(n) = T(n-1) + O(n) \longrightarrow T(n)$ is $\Theta(n^2)$

RANDOMIZED-SELECT: worst case still $\Theta(n^2)$ expected comparisons $< 4n$

RANDOMIZED-QUICKSORT: uses RANDOMIZED-PARTITION worst case $\Theta(n^2)$ expected $\Theta(n \log n)$

exactly 3 terms/clause; no variable appears more than once per clause

MAX-3-SAT: given 3-CNF φ, find assignment that maximizes # of true clauses

decision problem is NP-complete

k = # of clauses

RANDOMIZED-APPROX-MAX-3-SAT(φ) returns an assignment satisfying $\geq \frac{7}{8}k$ clauses (an $\frac{8}{7}$-approximation)

$\frac{7}{8}k \leq$ max # satisfied clauses $\leq k$

```
repeat          # iterations unbounded, but E[iterations] = polynomial
    for i=1 to n
        if random() < 1/2
            x_i ← T
        else
            x_i ← F
    result ← count-T-clauses(φ, x_1, ..., x_n)      poly
until result ≥ 7/8 k
return x_1, ..., x_n
```

expected polynomial time

RANDOMIZED-APPROX-MAX-3-SAT(φ)
repeat
    assign $x_1, \dots, x_n$ uniformly randomly
    result $\leftarrow$ count-T-clauses($\varphi, x_1, \dots, x_n$)
until result $\geq \frac{7}{8}k$
return $x_1, \dots, x_n$

let $p = P$(random assignment satisfies $\geq \frac{7}{8}k$ clauses)   then $p \geq \frac{1}{8k}$

$$E[\text{iterations}] = \sum_{j=1}^{\infty} j \cdot P(\text{stopped after exactly } j \text{ iterations})$$

$$= \sum_{j=1}^{\infty} j \cdot P(\text{didn't stop after } 1^{st}\ j-1) \cdot P(\text{stopped after } j)$$

$$= \sum_{j=1}^{\infty} j \cdot (1-p)^{j-1} \cdot p$$

$$= p \cdot \sum_{j=1}^{\infty} j \cdot (1-p)^{j-1} \qquad \left(\sum_{j=1}^{\infty} j \cdot r^{j-1} = \frac{1}{(1-r)^2} \text{ for } 0<r<1\right)$$

$$= p \cdot \frac{1}{p^2} = \frac{1}{p}$$

$$\leq 8k \qquad \left(p \geq \frac{1}{8k}\right)$$

RANDOMIZED-APPROX-MAX-3-SAT(φ)

repeat
    assign $x_1, \dots, x_n$ uniformly randomly
    result $\leftarrow$ count-T-clauses($\varphi, x_1, \dots, x_n$)
until result $\geq \frac{7}{8}k$
return $x_1, \dots, x_n$

let $p = P$(random assignment satisfies $\geq \frac{7}{8}k$ clauses)

let Z = # clauses satisfied $Z_i = \begin{cases} 1 & \text{if clause } i \text{ satisfied} \\ 0 & \text{otherwise} \end{cases}$ $E[Z_i] = \frac{7}{8}$

$P(\text{satisfied})$
$= 1 - P(\text{not satisfied})$
$= 1 - P(F \vee F \vee F)$
$= 1 - (\frac{1}{2})^3 = \frac{7}{8}$

$Z = Z_1 + \cdots + Z_k$ so $E[Z] = E[Z_1] + \cdots + E[Z_k] = \frac{7}{8}k$

also $E[Z] = \sum_{j=0}^{k} j \cdot \underline{P(j \text{ clauses satisfied})}$, call this p_j

$$= \sum_{j < \frac{7}{8}k} j \cdot p_j + \sum_{j \geq \frac{7}{8}k} j \cdot p_j$$ (split the sum)

$$\leq \sum_{j < \frac{7}{8}k} \left(\lceil \tfrac{7}{8}k \rceil - 1\right) \cdot p_j + \sum_{j \geq \frac{7}{8}k} k \cdot p_j$$ ($j \leq \lceil \frac{7}{8}k \rceil - 1$ for $j < \frac{7}{8}k$; $j \leq k$ otherwise)

$$= \left(\lceil \tfrac{7}{8}k \rceil - 1\right) \sum_{j < \frac{7}{8}k} p_j + k \cdot \sum_{j \geq \frac{7}{8}k} p_j$$ (factor out constants from Σ)

$$= \left(\lceil \tfrac{7}{8}k \rceil - 1\right) \cdot (1-p) + k \cdot p$$ ($p = \text{Prob}(\geq \frac{7}{8} \text{ clauses sat}) = \sum_{j \geq \frac{7}{8}k} p_j$)

$$\leq \lceil \tfrac{7}{8}k \rceil - 1 + kp$$ ($1 - p \leq 1$)

$$\tfrac{7}{8}k \leq \lceil \tfrac{7}{8}k \rceil - 1 + kp$$ ($E[Z] = \frac{7}{8}$)

$$kp \geq \tfrac{7}{8}k - \lceil \tfrac{7}{8}k \rceil + 1$$ (rearrange)

$$\geq \tfrac{1}{8}$$ ($\frac{7}{8}k - \lceil \frac{7}{8}k \rceil \geq -\frac{7}{8}$)

$$p \geq \frac{1}{8k}$$