Randomized Algorithms

For the average-case analysis of an algorithm, we need to assume some probability distribution on the space of all input instances of the problem

For sorting, we assume that all $n!$ permutations of n numbers are equally likely — we are on a shaky ground in assuming a particular distribution.

A different approach: randomized algorithm (vs. deterministic algorithm)

- do not assume about the distribution of instances
- incorporate randomization into the algorithm itself.

Las-Vegas Algorithms

Given I to P, a Las-Vegas algorithm uses some (r) random numbers, but except for choosing random numbers it proceeds completely deterministically.

LV's solution is always correct as in deterministic alg.

We say that LV solves P in expected time $T(n)$ if for *every* I such that $|I| = n$, LV solves I in expected time $\leq T(n)$.

By expected time we mean the average of all solution times of I by LV for all possible choice sequences of r random numbers (which we assume to be equally likely). Note the difference in assumption from average-case analysis.

Monte-Carlo Algorithms

A Monte-Carlo algorithm may produce an incorrect solution.

Let $e > 0$. We say that MC solves P with confidence greater than $1 - e$ if for *every* I the probability that MC will produce an incorrect solution is $\leq e$.

Randomized Quicksort

```
Quicksort(A,p,r)
  if p < r then
    q = Partition(A,p,r)
    Quicksort(A,p,q-1)
    Quicksort(A,q+1,r)
  fi
```

Partition(A, p, r)

1. Select a pivot element x ($A[p]$ in the original quicksort; a random element of $A[p..r]$ in the randomized quicksort).
2. All elements in $A[p..q-1]$ are $\leq x$.
3. All elements in $A[q+1..r]$ are $\geq x$.
4. The pivot element x is placed in $A[q]$.

The expected time $T(n)$ of Randomized Quicksort is $O(n \log n)$.

Randomized Selection

Problem: Given an array $A[1..n]$ and i, find the ith smallest element.

As in Quicksort, partition the input array recursively. Selection works only on one side of the partition.

```
Select(A,p,r,i)
  if p = r then return A[p] fi
  q = Partition(A,p,r)
  k = q - p + 1
  if i = k then return A[q]
  else if i < k then return Select(A,p,q-1,i)
  else return Select(A,q+1,r,i-k) fi
```

The expected time of Randomized Select is $O(n)$.

Verification of Polynomial Identities

Let $p(x_1, \ldots, x_n)$ be a polynomial in variables $x_1, \ldots, x_n$ over an arbitrary field F. The degree of p, denoted by $\deg(p)$, is $\max(i_1 + \cdots + i_n)$ over all multinomials $x_1^{i_1} \cdots x_n^{i_n}$.

Problem: verify whether a multivariate polynomial is identically zero.

Example: Given a matrix X containing $x_1, \ldots, x_n$, $det(X)$ is a polynomial in variables $x_1, \ldots, x_n$.

- Straightforward method: expand the polynomial into the sum of multinomials and check whether all coefficients are zero. But it takes lots of operations.
- Monte-Carlo algorithm: take a random point over a finite set I and evaluate the polynomial at the point.

Theorem 1 *Let $p(x_1, \ldots, x_n)$ be a polynomial in variables $x_1, \ldots, x_n$ over a field F such that p is not identically zero. Let I be any finite subset of F. Then the number of elements in I^n which are zeros of p is at most $|I|^{n-1}\deg(p)$.*

Proof. By induction on n. When $n = 1$, the number of zeros of polynomial p is at most $\deg(p)$. Thus the number of zeros in I is at most $\deg(p)$.

Assume the theorem holds for all polynomials with at most $n - 1$ variables. Let d be the degree of x_1 in $p(x_1, \ldots, x_n)$. We have $p(x_1, \ldots, x_n) = x_1^d q(x_2, \ldots, x_n) + r(x_1, \ldots, x_n)$ for some polynomials q, r. Let $(a_1, \ldots, a_n) \in I^n$ be a zero of p. There are two types of zeros of p.

- If $q(a_2, \ldots, a_n) = 0$, then p can be equal to zero for all $x_1 \in I$ (when r is identically zero). The total number of such zeros is at most $|I|(|I|^{n-2}\deg(q))$ since q has at most $|I|^{n-2}\deg(q))$

zeros by induction hypothesis.

- If $q(a_2, \ldots, a_n) \neq 0$, for each of such tuple $(a_2, \ldots, a_n)$, p is of degree d in x_1. So p has at most d zeros in I. Considering all tuples, there are at most $|I|^{n-1}d$ such zeros.

Therefore, the total number of zeros in I^n is $\leq |I|^{n-1}(d + \deg(q)) \leq |I|^{n-1}\deg(p)$. $\square$

Example: Let I be a finite subset of F containing 0, and $p = x_1 \cdots x_n$. The number of zeros is the number of all tuples minus the number of tuples without 0, i.e., $|I|^n - (|I| - 1)^n$. Since $(|I| - 1)^n = \sum_{i=0}^n \binom{n}{i}(-1)^i |I|^{n-i}$, the number of zeros is

$$\sum_{i=1}^{n} \binom{n}{i} (-1)^{i+1} |I|^{n-i} \leq n|I|^{n-1}.$$

Corollary 1 *Let $p(x_1, \ldots, x_n) \neq 0$ and I be a finite subset of F. The probability that a random tuple $(a_1, \ldots, a_n) \in I^n$ is a zero of p is $\leq \deg(p)/|I|$.*

Algorithm

1. Choose a finite subset of F whose size is at least $2\deg(p)$.
2. Select a random tuple v from I^n.
3. Evaluate p at v. If $p(v) \neq 0$, clearly p is not identically zero. Otherwise, declare that p is identically zero.

The error probability of such a method is $\leq 1/2$. Repeat the experiment k times. If $p(v) \neq 0$ at least once, p is not identically zero. Otherwise declare p is identically zero. The error probability is $\leq 1/2^k$.

Choosing a Large Number

Problem: Given n numbers, find a number that is a median or larger.

- Straightforward method: find a median by Deterministic Select or Randomized Select (LV algorithm). $O(n)$ time.
- Monte-Carlo algorithm: choose k numbers randomly, and return their maximum.

The error probability is $\leq 1/2^k$.