Algorithms 2005

Ramesh Hariharan

An Example: Bit Sequence Identity Check

- A and B have a sequence of n bits each (call these a and b).

- How do they decide whether their bit sequences are identical or not *without exchanging the entire sequences*?

Bit Sequence Identity Check

- Treat each bit string as a decimal number of size up to 2^n
- A chooses a random prime number p in the range $n^2..2n^2$ and sends it to B
- A and B takes their numbers modulo p and send the results to each other.
- The two numbers are equal only if the two remainders are equal.

Bit Sequence Identity Check

- False Positive: a!= b *but* a ´ b (mod p)
- False Negative: a = b *but* a !´ b (mod p)

- False negatives are not possible
- False positives are possible
  - How many primes in the range $n^2..2n^2$ will cause a false positive? (X)
  - How many primes are there in the range $n^2..2n^2$? (Y)
  - Probability of failure = X/Y

Bit Sequence Identity Check

- How many primes divide a-b? At most 2 * n/log n (Why?).
- So X<= 2 * n/log n.

- How many primes are there in the range $n^2..2n^2$?
  At least $n^2/2\log n$ (The Prime Number Theorem)

- So Y>= $n^2/2\log n$.

- Probability of failure = X/Y <= 4/n
- Number of bits exchanged = O(log n)

Bit Sequence Identity Check

Questions

- Why choose primes?
- How can one increase success probability even further?
- Can you show that n has at most O(log n/loglog n) primes?

Exercise
Polynomial Identity Checking

Given polynomials f(x) and g(x) of degree k each as black-boxes, can you determine if f(x) and g(x) are identical or not?

Randomized QuickSort

Each item is equally likely to be the pivot.

How fast does this run?

With high probability, in O(nlog n) time. Proof?

Random Variables

- Toss a coin which yields 1 with probability p and 0 with probability 1-p

- Probability Distribution, Random Variables

$$X = \begin{cases} 1 & p \\ 0 & 1-p \end{cases}$$

Mean, Variance

- Mean or E(X) = $1*p + 0 * (1-p) = p$

- Var(X) = $E((X-E(X))^2)$

  $= (1-p)^2*p + (0-p)\hat{}2*(1-p) = p(1-p)$

Independence

- Consider two coin toss outcomes represented by RV's X and Y
- X= 1 .5, 0 .5 Y= 1 .5,0 .5
- What is the joint distribution of X and Y?

Independent		**Dependent**
1 1 .25		1 1 .5
1 0 .25		0 0 .5
0 1 .25		
0 0 .25		

For independence,

Pr(X|Y)=Pr(X)

Pr(X=0/1 and Y=0/1) = Prob(X=0/1) Prob(Y=0/1)

Independence

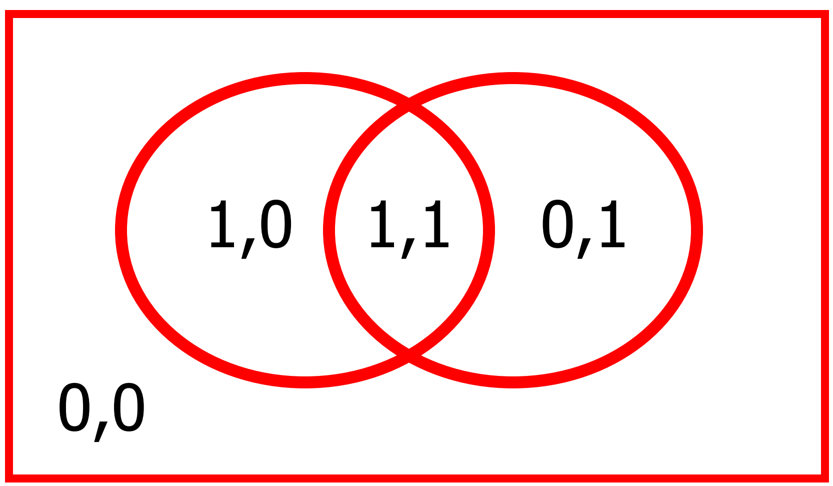
$Pr(X=0/1 \text{ and } Y=0/1) = Prob(X=0/1)\, Prob(Y=0/1)$

- $E(XY)=E(X)E(Y)$ if X and Y are independent
- $E(X+Y)=E(X)+E(Y)$ always

- $Var(X+Y)=Var(X)+Var(Y)$ if X and Y are independent

Union Bound and Mutual Exclusion

- Pr(X=1 or Y=1) = Pr(X=1) + Pr(Y=1)-Pr(X=1 and Y=1)
- Pr(X=1 or Y=1) <= Pr(X=1) + Pr(Y=1)
- Pr(X=1 or Y=1) = Pr(X=1) + Pr(Y=1) under mutual exclusion



A Coin Tossing Problem

- If we toss a fair coin repeatedly and independently, how many tosses need to be made before we get i heads. Let X be this random variable

- Pr(X=k) = [k-1 C i-1] / 2^k (Why?Is independence used?)

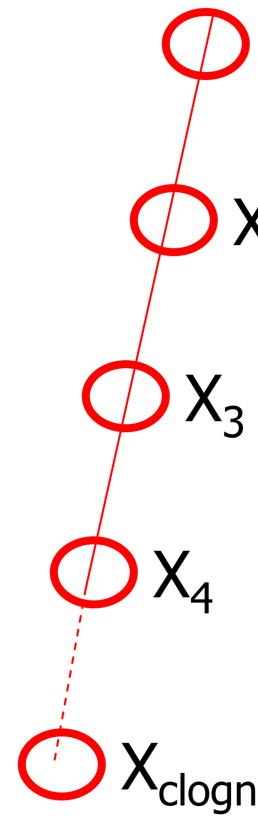
  <= $(ek/i)^i/2^k$ (Why?)

- For i=log n and k=clog n,

  Pr(X=k) <= $1/n^2$

Randomized QuickSort

- Consider a particular path
  - $X_i = 1$, if the size reduces by 3/4ths or more at the ith node in this path; this happens with prob .5
  - $X_i = 0$, otherwise, with probability .5
- There can be at most log n i's for which $X_i=1$

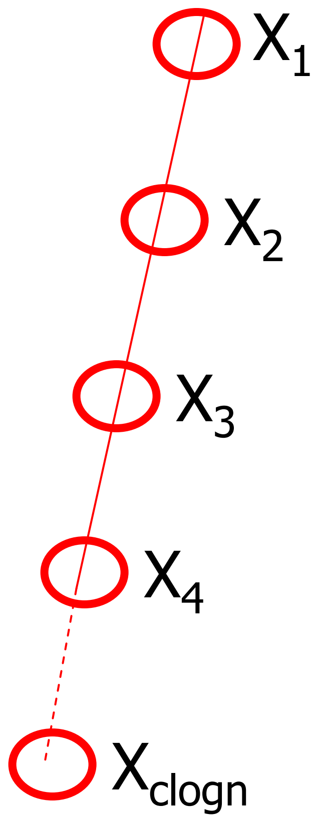


How many coin tosses are needed to get log n heads? The length of the path L is bounded by this number.

$$\Pr(L = c\log n) < 1/n^2$$

Randomized QuickSort

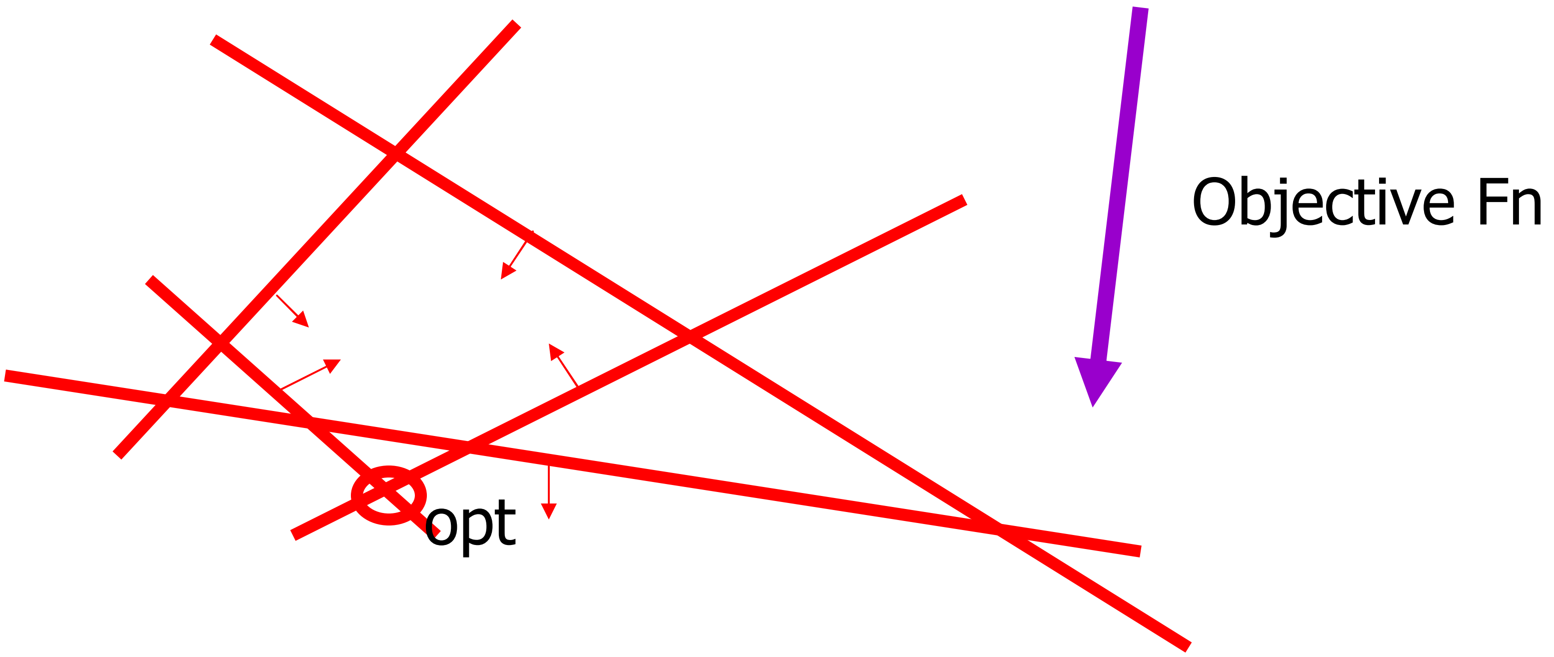
- Pr(L=4log n)<1/n^2 for a particular path
- But we need it to be small for all possible paths
- There are only n paths
- Use the union bound
- Pr(L_1=4log n or L_2=4log n or L_3=4log n… L_n=4log n)< 1/n
- Overall: O(nlog n) time with probability at least 1-1/n



QuickSort Puzzle

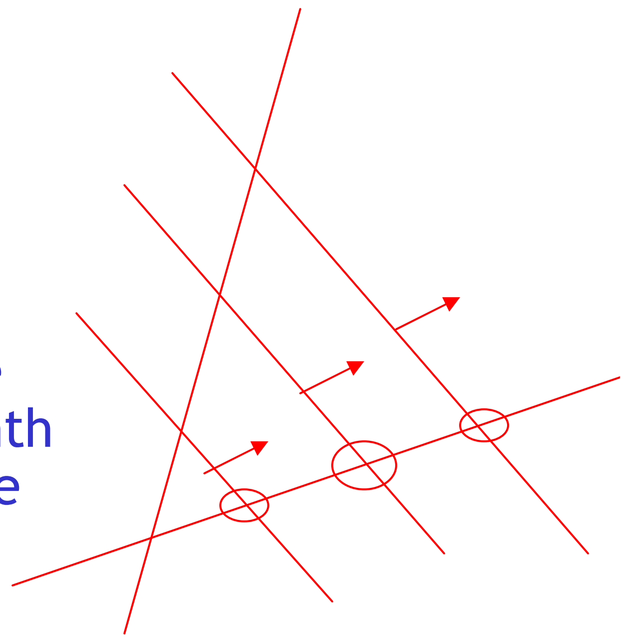
- In a spreadsheet, clicks on a column header sort the data in ascending and descending order alternately.
- Two clicks on the column header caused the program to crash. Why?

2D Linear Programming



2D Linear Programming

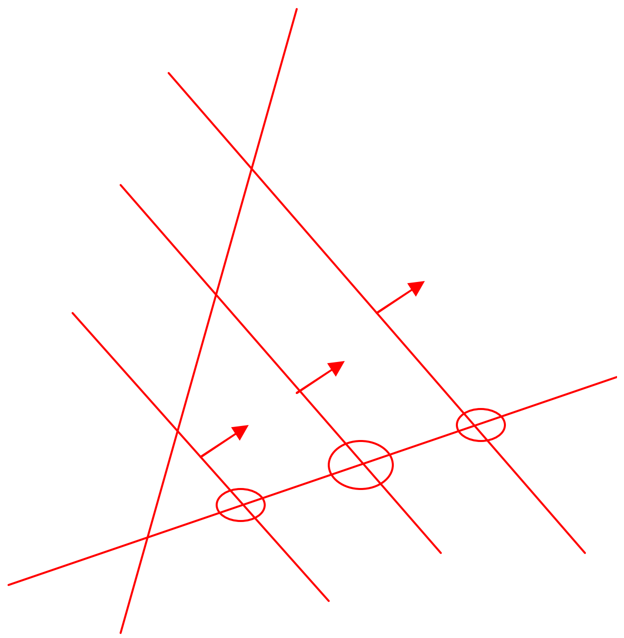
- Assume that the feasible region in non empty
- Find optimum for n-1 constraints recursively
- Add the nth constraint;
- Check if the optimum changes, if so compute the new optimum by finding the intersection of the nth constraint with all previous constraints: O(n) time
- How often does the optimum change?
- Total time is $O(n^2)$



2D Linear Programming

Randomized Algorithm

- Consider constraints in a random order
- In the example, how many times does the maximum change?
- In a randomly ordered sequence, if you compute max from left to right, how many times does the max variable get updated?



2D Linear Programming

What Happens in General

- $X_i = i$ if the optimum changes when the ith constraint is added
- $X_i = 1$ otherwise
- total time $T = \sum X_i$,
  - $E(T) = \sum E(X_i)$ *Linearity of Expectation*
  - $Pr(X_i = i) = 2/i$ *Why*
  - $E(X_i) = 2/i * i + 1-2/i <= 3$
  - $E(T) <= 3n$

2D Linear Programming

- Consider X_i for a fixed choice of the first i hyperplanes
  (i.e., the set H of first i hyperplanes is fixed but not their relative order)
- Suppose we calculate E(X_i|H)
- How do we recover E(X_i) from this?

2D Linear Programming

Determining $E(X_i|H)$

- Given H is fixed, the optimum over H is fixed even though the order of hyperplane addition in H may vary.
- This optimum lies on at least 2 hyperplanes.
- The probability that the last addition will cause a change in optimum is at most $2/i$.

The Random Walk Problem

- Start at the origin and take a step in either direction with probability .5 each; repeat n times. How far are you from the origin?

- X_i = +1 w.p .5
- X_i = -1 w.p .5
- Assume X_is are independent
- $X = \sum X_i$
- $E(X) = \sum E(X_i) = 0$

- Does this mean you will be at the origin after n steps?

Expectation vs High Probability

- Can an expected bound be converted to a high probability bound?
- We want a statement of the following kind:
  - The time taken is O(n) with probability at least .9
  - After n steps, we will be between x and y with probability at least .9

Tail Bounds

Prove these Bounds

- Markov's

  Pr(X>k)<E(x)/k, for positive RV X

- Chebyschev's

  $Pr((X-E(X))^2>k)<Var(x)/k$, for all RV X

Tail Bounds for Random Walk

- Markov's: Does not apply due to non-positivity
- Chebyschev's

  $Pr((X-0)^2>cn)<n/cn$

  $Pr(|X|>sqrt(cn))<1/c$

So with high probability, one is within $\Theta(sqrt(cn))$ from the center.

Multiple Random Walks

- Assume n random walkers
- After n steps, how far is is the furthest walker from the origin?
- We can use the union bound; the probability that at least one of the walkers is distance c away is at most n times the probability that a specific walker is distance c away: this comes to n * n/c^2 using Chebyschev's bound.
- This does not give us anything useful.
- Is there a sharper bound?

Chernoff's Bound

- With what probability does the sum of independent RVs deviate substantially from the mean?

  - RVs $X_1..X_n$,
  - Independent
  - X_i has mean m_i
  - X_i's are all between -M and M

Chernoff's Bound

- $Pr(\sum (X_i - m_i) > c)$

$=$	$Pr(t \sum (X_i-m_i) > t\, c)$	$t>0$
$=$	$Pr(e^{t\sum (X_i-m_i)} > e^{tc})$	raise to e
$<=$	$E(e^{t\sum (X_i-m_i)}) / e^{tc}$	Markov's
$=$	$\prod E(e^{t (X_i-m_i)}) / e^{tc}$	Independence
$<=$	$\prod (.5 (1- m_i/M) e^{t (-M-m_i)} + .5 (1+ m_i/M) e^{t (M-m_i)}) / e^{tc}$	Convexity(prove this)
$<=$	$\prod (.5\, e^{t(-M-m_i)-m_i/M} + .5\, e^{t(M-m_i)+m_i/M}) / e^{tc}$	$1+x<=e^x$
$=$	$\prod e^{-tm_i} \prod (.5\, e^{-tM-m_i/M} + .5\, e^{tM+m_i/M}) / e^{tc}$	e^{-tm_i} common
$<=$	$e^{-t\sum m_i + \sum .5(tM+m_i/M)^2 - tc}$	$.5(e^x + e^{-x}) <= e^{x^* x/2}$
$<=$	$e^{\sum .5t^2M^2 + \sum .5(m_i/M)^2 - tc}$	open up the square
$<=$	$e^{-.5c^2/\sum M^2 + \sum .5(m_i/M)^2}$	optimize for t
$<=$	$e^{-.5c^2/n.M^2 + \sum .5(m_i/M)^2}$	
$<=$	$e^{-(c^2/n- \sum m_i^2)/2M^2}$	

Multiple Random Walks

- Assume n random walkers
- After n steps, how far is is the furthest walker from the origin?
- We can use the union bound; the probability that at least one of the walkers is distance c away is at most n times the probability that a specific walker is distance c away:
- Using m_i=0, M=1, c=sqrt(4nlog n) in the Chernoff Bound, we get that the above probability is n * $1/n^2$ = 1/n

Exercises

- Generalize to X_is between A and B
- Generalize to $Pr(\sum (X_i - m_i) < -c)$ for $c>0$
- Use in the Chernoff Bound to show the bound obtained earlier on the coin tossing problem used in the QuickSort context

Exercises

- Consider a linked list in which each node tosses an independent coin (heads with p tails with 1-p). Bound the largest inter-head distance.

- Throw n balls into n bins, each ball is thrown independently and uniformly. Bound the max number of balls in a bin

- Also see Motwani and Raghavan

Exercise on Delaunay Triangulation

- Insert points in a random order
- Suppose n-1 points have been inserted and a triangulation computed
- Add the nth point and locate the triangle it is contained in (assume it is contained in a unique triangle and is not sitting on an edge)
- What processing do you do and how long does it take?

Facts on Delaunay Triangulation

- Voronoi Diagram: Decompose the plane into cells, a cell comprising all locations which are closest to a specific point. There is one cell per point.
- Delaunay: Dual of Voronoi, cells become points, adjacent cells(points) are connected by lines.
- The Delaunay graph is planar
- A triangulation is a delaunay triangulation if and only if the circumcircle of any triangle does not contain a point in its strict interior.
- An edge in a delaunay triangulation if and only if there exists a circle which passes through the endpoints of this edge but does not contain any other points in its strict interior.

Thank You