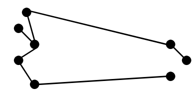
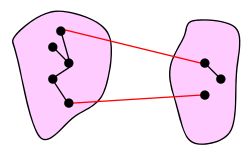
Lecture 6, Oct. 8, 2009

Power of randomization

- Given undirected graph G



- Divide set of nodes V into 2 parts, such that at least half of the edges connect a node in one part to a node in another part.
- Red edges are "crossing"
- Wrong solution:

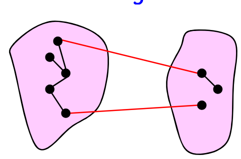


- Are we guaranteed that there is a solution at all ??

64

More about partitioning

- Algorithm: Assign each node to right/left with probability 50%

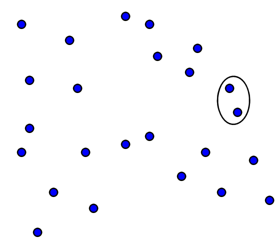


- Analysis:
  - » Probability that an edge crosses is 50%
  - » Expected number of crossing edges = $\sum$ over all edges of (1 x Pr[edge is crossing])=|E|/2
- But this implies that a solution exists with at least half crossing !

65

Closest pair of point

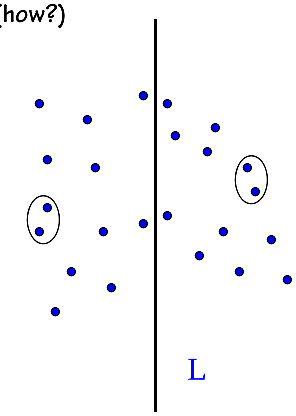
- Given points $\{(x_i, y_i)\ i=1\ldots n\}$ on a plane
- Goal: find two points that are closest to each other
- Obvious solution $\Theta(n^2)$ (how?)
- Can we do better ?



66

Divide and conquer approach

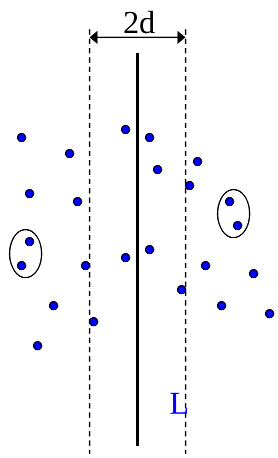
- Divide plane using vertical line L into 2 parts (how?)
- Recursively find closest pair on the right
- Recursively find closest pair on the left
- How to combine the results ?



67

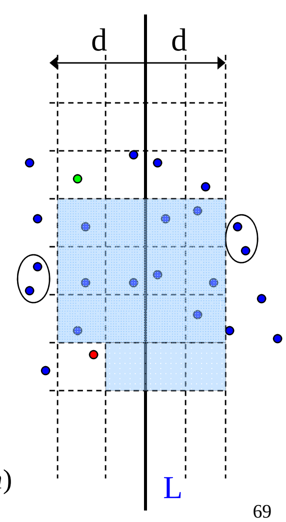
Combining the results

- Let d be minimum distance between points on the right and on the left.
- Observe that there might be 2 points, one on right one on left, that are closer than d
- Claim: any such points reside in a “band” of d around L !
- Why ??
- Does this help us with running time ?
- What if all points are in this band ?



Combining the results

- Focus only on points in the “band”
- Sort points by y coordinate
- Place a grid with d/2 side onto the band
- At most one point per cell ! (Why ??)
- Need to look “up” only 15 points (going by increasing y coordinate)



$$T(n) = 2T(n/2) + \Theta(n) \quad \Rightarrow \quad T(n) = \Theta(n \log n)$$

Order Statistics

- Problem: Find the *i*-th smallest element (Rank-*i*).
  If $i=1$ ⟶ Minimum
  $i=n$ ⟶ Maximum
  $i=n/2$ ⟶ Median
- Possible solution:
  - » Sort
  - » Index into A(i).

  } $O(n \lg n)$
- We can do better ! (Can we hope to do better than linear time ?)

70

Randomized selection

- Divide and conquer approach:

```
RS(A,p,r,i)
  if p==r  then return A(p)
  q=RandomPartition(A,p,r)
  k=q-p+1
  if i< k  then return RS(A,p,q-1,i)
     i>k   then return RS(A,q+1,r,i-k)
     i==k  then return A(q)
```

- Correctness:
  - » Assume correct for size at most n=r-p+1
  - » after the partition, the arrays are smaller than n, can apply induction.
  - » Claim: need to search only one part
  - » Need to prove each one of the 3 cases.

71

Performance of Random Selection

- Lucky case: $T(n) \le T\left(\frac{9}{10}n\right) + \Theta(n)$

$$\left.\begin{array}{l} n^{\lg_b a} = 1 \\ \text{check } 0 < c < 1: af(n/b) \le cf(n) \end{array}\right\} \Rightarrow \; case\ 3 \;\Rightarrow\; T(n) = O(n)$$

- What if 99/100 instead of 9/10 ??

- Bad case:

$$T(n) = T(n-1) + \Theta(n) \;\Rightarrow\; T(n) = \Theta(n^2)$$

72

Analysis continued

- Let T(n) be the expected running time.
  Condition on partition outcome:

$$T(n) = E_{partition\_outcome}\left[\text{Expected time conditioned on last partition outcome}\right]$$

$$\le \frac{1}{n}\sum_{k=0}^{n-1} T(\max\{k, n-k-1\}) + \Theta(n)$$

$$\le \frac{2}{n}\sum_{k=\lfloor n/2 \rfloor}^{n-1} T(k) + \Theta(n)$$

73

Analysis continued

- Substitute $T(n) \le cn$, choose c large enough for T(1):

$$T(n) \le \frac{2c}{n} \sum_{k=\lfloor n/2 \rfloor}^{n-1} k + \Theta(n)$$
$$\le \frac{2c}{n}\left[\sum_{k=1}^{n-1} k - \sum_{k=1}^{\lfloor n/2 \rfloor -1} k\right] + \Theta(n)$$
$$\le \frac{2c}{n}\left[\frac{n(n-1)}{2} - \frac{(n/2-2)(n/2-1)}{2}\right] + \Theta(n)$$
$$= cn + \underbrace{(const_1 + \Theta(n) - const_2 \cdot c \cdot n)}_{\le 0 \text{ for large enough c}}$$

$$\Rightarrow \quad T(n) = O(n)$$

Can we claim $\Omega(n)$?

74

Deterministic Order Statistics

- The randomized order statistics is very fast in practice (just like quick-sort, same additional tricks will help).
- Theoretically interesting question: Is there a deterministic linear time order-statistics algorithm ?
- Deterministic selection algorithm (select *i*-th smallest):
  - » Divide n elements into groups of 5.
  - » find median in each group (brute force)
  - » Use select recursively to find median among n/5 medians. (i.e. select n/2-nd smallest)
  - » Partition around this median.
  - » Recurse on the "appropriate" part, update *i* if necessary.

75

Deterministic order statistics -cont

- **Correctness - as before. All we changed was the pivot choice.**
- **Time:** half of the medians are $\le x \Rightarrow \frac{1}{2}\left\lfloor \frac{n}{5} \right\rfloor \ge \left\lfloor \frac{n}{10} \right\rfloor$

  each median brings 3 elements $\Rightarrow 3\left\lfloor \frac{n}{10} \right\rfloor$ total

  for $n \ge 50$, we have $3\left\lfloor \frac{n}{10} \right\rfloor \ge \frac{n}{4} \Rightarrow$ at least $n/4$ elements are $\le x$.

  Similarly, at least $n/4$ elements are $\ge x$.

> $n = 10x + y,\ y \le 9$
>
> $3\left\lfloor \frac{n}{10} \right\rfloor = 3x, \quad \frac{n}{4} = 2.5x + \frac{y}{4} \le 2.5x + \frac{9}{4}$
>
> so now we solve for: $3x \ge 2.5x + \frac{9}{4}$, x integer
>
> $\Rightarrow \quad x \ge 5$

76

Deterministic order statistics -cont

- **Recursion:**

$T(n) \le T(n/5) + T(3n/4) + \Theta(n)$

Substitution: $T(n) \le cn$,
choose c large enough to cover $T(1)$.

$T(n) \le cn/5 + 3cn/4 + \Theta(n)$
$\quad = cn + \underbrace{(\Theta(n) - cn/20)}_{<0 \text{ for large enough c}}$

$\Rightarrow \quad T(n) = O(n)$

In fact, we have $T(n) = \Theta(n)$
(*Why*??)

77

Deterministic selection

- Homework:
  analyze with groups of 4 elements and groups of 6 elements.
  What if groups are n/10 elements each ?

- Observe that we can get deterministic variant of quicksort !
  - Can use as a black-box O(n) partitioning into 2 equal parts.
  - We get recurrence $T(n)=2T(n/2)+\Theta(n)$, giving us $\Theta(n \lg n)$ total running time.
  - (do you think it will work well in practice ??)

78