CPSC 490 – Problem Solving in Computer Science

Lecture 21: Binary Search & Ternary Search

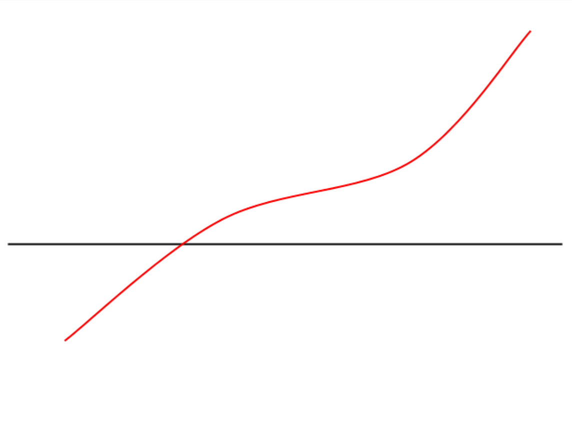
Jason Chiu and Raunak Kumar

2017/03/15

University of British Columbia

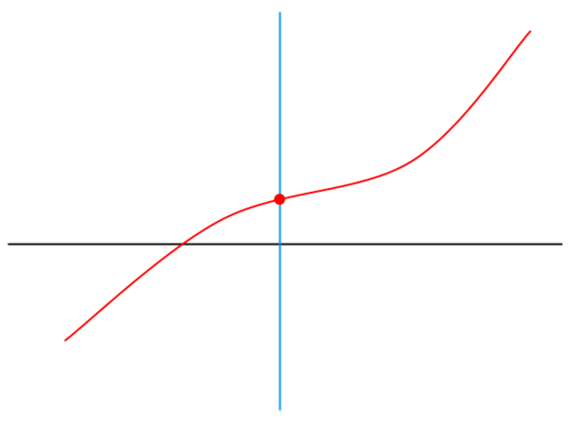
Binary Search

Suppose we want to find when a function crosses zero



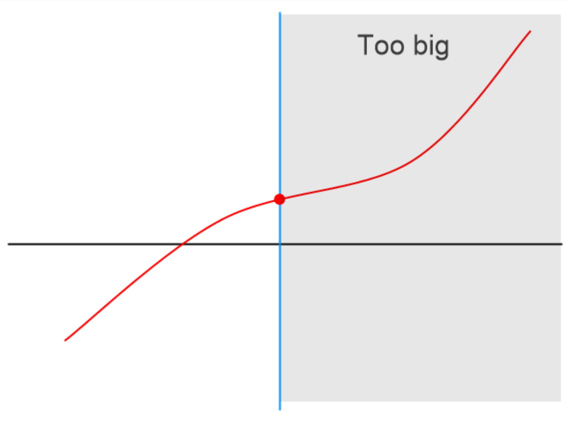
Binary Search

Evaluate it at any point...



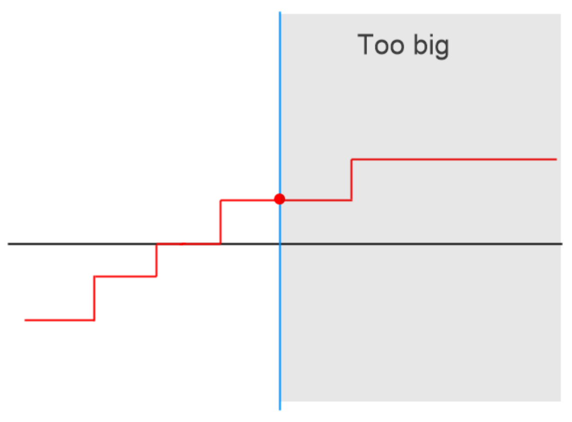
Binary Search

If too big, then kill right half! Every step cuts search space in half!



Binary Search

Works with non-strictly increasing functions too!



Binary Search (Continuous)

Find smallest zero of a non-decreasing function $f : \mathbb{R} \to \mathbb{R}$

```
BINARY_SEARCH(lb, rb, f):
  while rb - lb > EPS:
    set mid = lb + (rb-lb)/2
    if f(mid) >= 0: rb = mid
    else: lb = mid
  return (lb+rb)/2
```

Time complexity: $O(\log \varepsilon)$

Binary Search (Discrete)

Find smallest zero of a non-decreasing function $f : \mathbb{Z} \to \mathbb{R}$

```
BINARY_SEARCH(lb, rb, f):
  if f(lb) >= 0: return lb
  while rb - lb > 1:
    set mid = lb + (rb-lb)/2
    if f(mid) >= 0: rb = mid
    else: lb = mid
  return rb
```

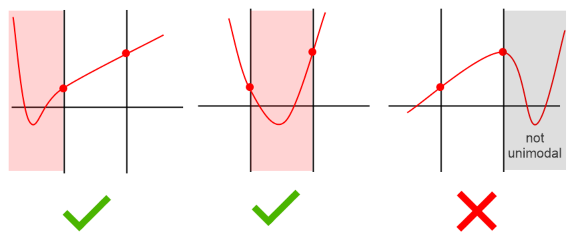
Time complexity: $O(\log n)$

Ternary Search

Can we use a similar strategy to minimize a unimodal function?

Can we use a similar strategy to minimize a unimodal function?

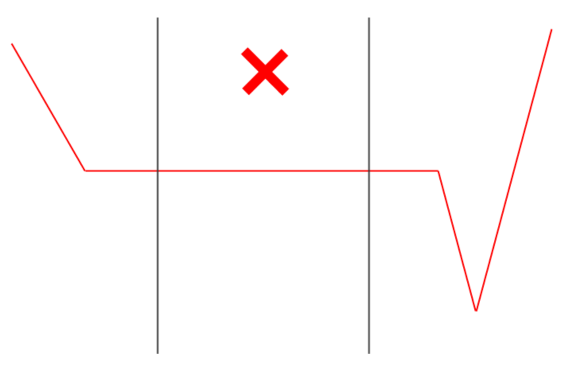
Evaluate it at two points in the middle, if left < right, then already increasing in middle section, so minimum not in right section!



Ternary Search

What if two probe points are equal?

Must be strictly unimodal (i.e. strictly decrease to minimum, then possibly stay flat at minimum, then strictly increase), to avoid failure!



Ternary Search

What if we put probe points closer to middle?

Ternary Search

What if we put probe points closer to middle?

- Converges faster! (by a constant factor)
- So choose $1/2 - \epsilon$ and $1/2 + \epsilon$?
  - This is just binary search on derivative
  - Make sure ϵ not too small, otherwise numerical error!

Ternary Search

What if we put probe points closer to middle?

- Converges faster! (by a constant factor)
- So choose $1/2 - \epsilon$ and $1/2 + \epsilon$?
  - This is just binary search on derivative
  - Make sure ϵ not too small, otherwise numerical error!

Can we use less probes per iteration?

Ternary Search

What if we put probe points closer to middle?

- Converges faster! (by a constant factor)
- So choose $1/2 - \epsilon$ and $1/2 + \epsilon$?
  - This is just binary search on derivative
  - Make sure ϵ not too small, otherwise numerical error!

Can we use less probes per iteration?

- Golden section search: probe at golden ratios
  $\Rightarrow$ one probe <u>reused</u> next iteration
- Let $d = R - L$, $\phi = (\sqrt{5} - 1)/2$, then probe at $L + \phi d$, $R - \phi d$

Ternary Search

What if we put probe points closer to middle?

- Converges faster! (by a constant factor)
- So choose $1/2 - \epsilon$ and $1/2 + \epsilon$?
  - This is just binary search on derivative
  - Make sure ϵ not too small, otherwise numerical error!

Can we use less probes per iteration?

- Golden section search: probe at golden ratios
  $\Rightarrow$ one probe <u>reused</u> next iteration
- Let $d = R - L$, $\phi = (\sqrt{5} - 1)/2$, then probe at $L + \phi d$, $R - \phi d$

Does it work in 2D/3D?

Ternary Search

What if we put probe points closer to middle?

- Converges faster! (by a constant factor)
- So choose $1/2 - \epsilon$ and $1/2 + \epsilon$?
  - This is just binary search on derivative
  - Make sure ϵ not too small, otherwise numerical error!

Can we use less probes per iteration?

- Golden section search: probe at golden ratios
  $\Rightarrow$ one probe reused next iteration
- Let $d = R - L$, $\phi = (\sqrt{5} - 1)/2$, then probe at $L + \phi d$, $R - \phi d$

Does it work in 2D/3D? Yes if f is convex.

Ternary Search

Find minimum of strictly unimodal function $f : \mathbb{R} \to \mathbb{R}$

```
TERNARY_SEARCH(lb, rb, f):
  while rb - lb > EPS:
    set mid1 = lb + (rb-lb)/3
    set mid2 = lb + 2*(rb-lb)/3
    if f(mid1) < f(mid2): rb = mid2
    else: lb = mid1
  return (lb+rb)/2
```

Time complexity: $O(\log \varepsilon)$

Problem 1: BFS

Given weighted graph $G = (V, E)$, find path from u to v that minimizes the second maximum weight edge along the path.

Problem 1: Solution

Binary search for the answer.

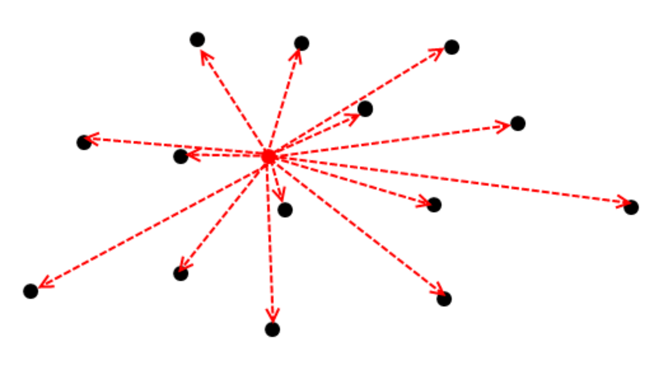
Suppose want 2nd maximum weight $\leq w$, then BFS allowing only 1 edge with weight $\geq w$ by keeping track of state (node, has edge $\geq w$).

How to get the path? When adding v from u, add "parent pointer" from v to u, then trace back parent pointers from destination.

Time complexity: $O((V + E) \log E)$

Problem 2: Convex Optimization

Find the point that minimizes the total distance to all the other points. Your solution must work over d dimensions ($d \leq 5$).



Problem 2: Convex Optimization

Observation 1: distance function is convex.

Observation 2: sum of convex functions is convex

$\Rightarrow$ do d-dimensional ternary search

- Ternary search on dimension 1
- Given a probe point in dimension 1, do $(d-1)$-dimensional ternary search to find minimum with dimension 1 fixed, take this as the "value" of the function

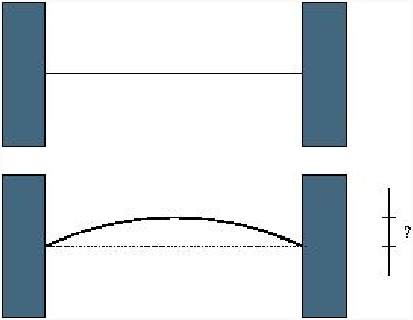
Time complexity: $O(n \log^d \varepsilon)$

Problem 3: Physics

When temperature increases by n degrees, the length of the rod expands and follows formula $L(n) = (1 + nC)L$.

If left and right end of rod is fixed, then it curves as it expands and the curve is an arc (i.e. part of a circle).

Given n, C, L find how much the center of the rod has moved.



Problem 3: Solution

If you try to solve the equation you see that you cannot get a nice closed formula so easy solution is to binary search – but on what?

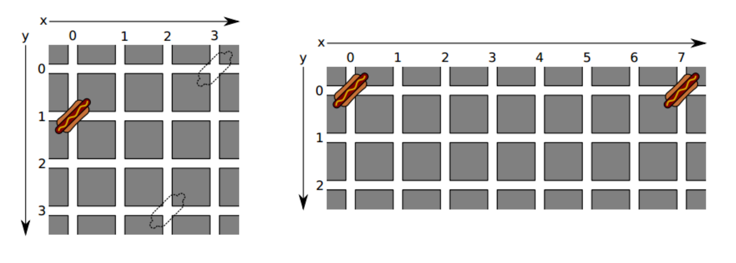
$\Rightarrow$ Pick a variable that is bounded. For example, binary search on the angle of the arc, not the radius of the circle.

Time complexity: $O(\log \varepsilon)$

Problem 4: Segment Tree / Binary-indexed Tree

In $n \times n$ street grid ($n \leq 1000$), there are already (up to) n hot dog stands. I want to open two new hot dogs stands, where should we put them? Hot dog stands must be on intersection.

Maximize the minimum Manhattan distance from either of the new hot dog stands to any other hot dog stands (include each other)!



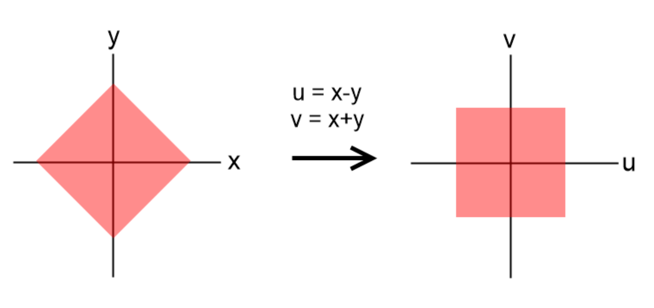
Source: BAPC 2012

Problem 4: Solution

Binary search for the maximum possible Manhattan distance!

"Exclusion zone" for an existing hot dog stand is a diamond
$\Rightarrow$ Rotate 45 degrees and it becomes a square!
This can be accomplished by the map $(x, y) \mapsto (x - y, x + y)$.

Observation: just as interval queries are "easy" in 1D, rectangle queries are "easy" in 2D



Problem 4: Solution

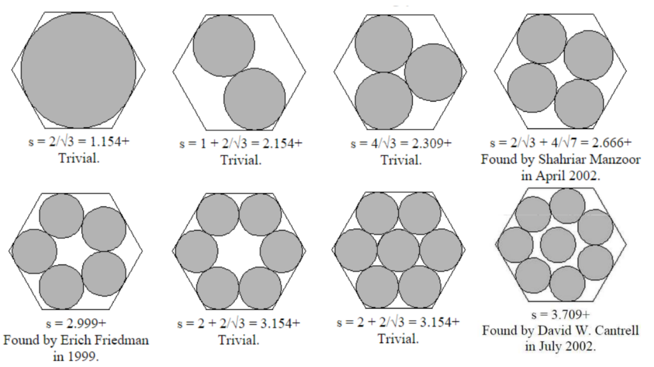
- BFS from all hot dog stands at once to compute distance from every point to nearest existing hot dog stand
- Rotate all points 45 degrees by the map $(x, y) \mapsto (x - y, x + y)$
- Binary search for the answer: to see if distance d possible
  - Find all the points with distance $\geq d$ from existing hot dog stand, say there are m of them
  - For every such point p, check if any points outside $2d \times 2d$ square centered at p, i.e. check if square has less than m points

Need a data structure for point insertion and range sum in 2D $\Rightarrow$ 2D Binary-indexed Tree is the easiest choice, but others are possible.

Time complexity: $O(n^2 \log n)$

Problem 5: Very Difficult Geometry

Find the minimum side length of a regular hexagon such that you can pack k unit circles inside without overlap. Constraint: $k \leq 8$.



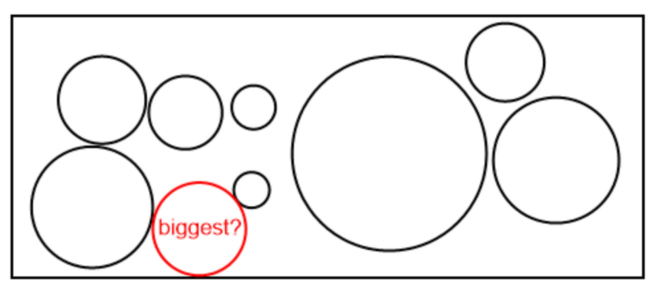
Problem 5: Solution

You could try to compute a complicated formula… or binary search!

Much easier to try to place circles in a hexagon rather than trying to enlarge a hexagon to fit circles.

Problem 6: Circle Packing Again

What is the biggest circle you can place in this rectangle that does not overlap any other circle or any side of the rectangle?



Problem 6: Solution

Binary search for the answer!

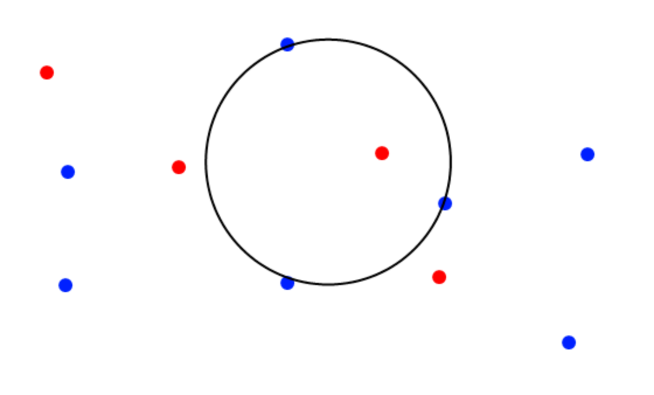
How to determine of radius r is possible?

- Enlarge all circles by radius r, shrink rectangle sides by r, check if new circles cover new rectangle
- Idea: if circles do cover entire rectangle, then every point on the perimeter is inside some the circle or outside rectangle
- $\Rightarrow$ for every circle, intersect with every other circle and every side of rectangle to get "angular interval" of intersection, compute if interval union is 2π.

Time complexity: $O(n^2 \log n \log \varepsilon)$

Problem 7: Angular Sweep

What's the biggest circle that contains at least 1 red point and no blue points?



Problem 7: Solution

Binary search for the answer!

How to determine of radius r is possible?

- Rotate a circle of radius r around every single point

Time complexity: $O(n^2 \log n \log \varepsilon)$

End of material for Assignment 5

Next Class

Wed: Assignment 2/3/4 Discussion

Fri: Maximum Flow