Colorado State University

CS 320 Fall 2021
Line of Sight

Sanjay Rajopadhye
Colorado State University

Announcement: Friday Exam

- Students in on-campus sections (001, 002) take the exam in the CSB 110 lab. In person proctoring.
  - Exam hours: 8:00AM to 4:00PM (the lab closes at 6:00PM, but it's a 2-hour exam)
- Online students (sec 801) take it online with Honorlock proctoring.
  - There will be a practice exam on Thursday
- The actual exam is a quiz in Canvas. There will be written questions
- We will grade the exam on Sunday.
  - If your score is high enough (above a TBD threshold) we will bump it up to 100
  - If your score is too low (below a TBD floor), you should really drop this class
  - If your score is in between, we will allow you one retake, and you will get the average of the two attempts
    - Retake on Tuesday, graded on Wednesday.

Colorado State University 2

Digression PA1 Algorithm

The problem statement:
Given

- an array, X[i,j] of the elevations of points in a (hilly) terrain, and
- information about where the sun currently is,
determine, for each point, whether it is sunlit or in the shade.

Also called the line-of-sight problem.

Imagine that you were positioned at the sun (beware Icarius) then which points in the hilly terrain would be in your line of sight and which would be hidden from view

Colorado State University 3

Concretely

Inputs:

- X[i, j] is an n x n array of (floating point) numbers (in meters)
- The angle of elevation of the sun $\Theta \leq 90°$
- The angle of azimuth of the sun, Φ
- The horizontal distance (in meters) between adjacent points, h, (the resolution or scale of our data)

Output:

- S[i, j] an n x n array of Booleans:
  - If [i, j] is in the shade, S[i, j] is 1
  - Otherwise it is 0

Simplifying assumptions & conventions:

- The azimuth is due west, $\Phi = 0$. So only points to the west (i.e., on the i^{th} row) can cast a shadow on [i, j]
- So, focus on just the i^{th} row of X, which we re-name as R, a 1-dimensional array (an outer loop iterates over each row). This Simplifies notation/figures on next few slides

Colorado State University 4

But first a digression

Some easy problems

- Add up n elements of an array $\Theta(n)$
- Max of all elements in an array $\Theta(n)$

```
r = 0
for i in range(length(X)):
  r += X[i]
```

```
r = 0; //minus infinity
for i in range(length(X)):
  r max= X[i]
```

What if you wanted all intermediate sums/maxima

$$Y[i] = \sum_{j=0}^{i} X[j]$$

```
for i in range(length(X)):
  Y[i] = 0
  for k in range(i)
    Y[i] += X[k]
```

Lower bound? $\Omega(n)$

First (direct) algorithm? $O(n^2)$

Can we do better? $O(n)$

```
Y[0] = X[0]
for i in range(length(X)-1):
   Y[i+1] = Y[i] + X[i+1]
```

Colorado State University 5

Back to LoS

Use predicate logic and some simple reasoning. And remember that we only look at the i^{th} row.

- A point at j is in the shade, if some point to its west casts a shadow in it, i.e.,

$$\exists k: 0 \le k < j\ , \quad \frac{R[k] - R[j]}{h(j-k)} > \tan\Theta$$

- First algorithm implements this as a loop (quadratic time per row)
- Second algorithm does an "early exit:" as soon as we find a point that puts j in the shade, we exit the loop
- Next, we improve the complexity using the idea of the running max. First change existential to universal

Colorado State University 6

Improvement

$$\neg\left(\forall k: 0 \le k < j, \quad \frac{R[k]-R[j]}{h(j-k)} \le \tan\Theta\right)$$

Calculate the negation: j is sunny if

$$\forall k: 0 \le k < j, \quad \frac{R[k]-R[j]}{h(j-k)} \le \tan\Theta$$

Take all terms involving j and k on opposite sides

$$\forall k: 0 \le k < j, R[j] + hj\tan\Theta \ge R[k] + hk\tan\Theta$$

LHS is independent of the quantified variable. Distribute it and use max

$$R[j] + hj\tan\Theta \ge \max_{0\le k<j} R[k] + hk\tan\Theta$$

Calculate the RHS using the running max idea

- Wim's slides 44 – 77

Common Running times