CS 260: Foundations of Data Science

Prof. Sara Mathieson

Spring 2025

HAVERFORD
COLLEGE

Admin

- Lab 5 released (partners optional), due after Spring Break
- Lab 3 grades will be up soon
- No class next Thursday (I'm at a conference)

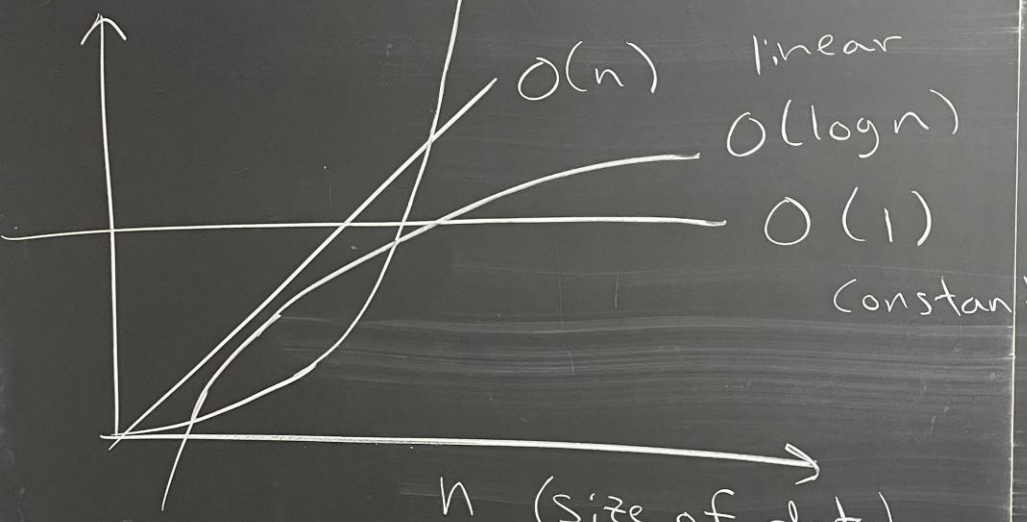
- **Midterm 1: next Tuesday in class**
  - Study sheet created by 1 (one page front and back)
  - No other notes or resources

Outline

- Review runtime
- Handout 12
- Finish algorithmic bias
- Discussion: admissions at Haverford

Outline

- Review runtime
- Handout 12
- Finish algorithmic bias
- Discussion: admissions at Haverford



$O(n) + O(\log n) \Rightarrow O(n)$

logarithmic

$O(cn) \Rightarrow O(n)$

c: constant

$O(5n + 11n) \Rightarrow O(n)$

$O(n^k + n^j) = O(n^{\max(k,j)})$

n^5 constants

$+ n^7 \Rightarrow O(n^7)$

1. work inside out (start with inner loop)
2. what is changing?
3. what is constant?
4. log (see halving)
5. exp (see doubling)

Outline

- Review runtime
- Handout 12
- Finish algorithmic bias
- Discussion: admissions at Haverford

① $O(n)$

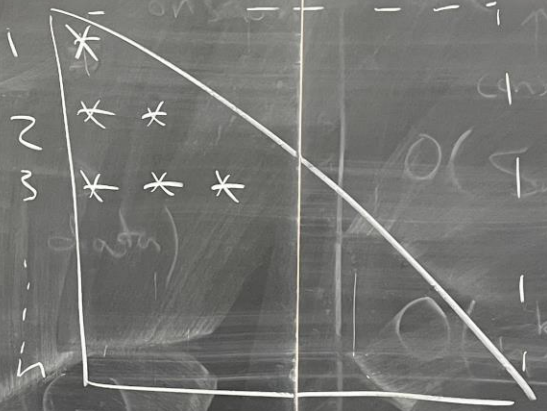
② $O(1)$

③ $O(n)$

④ $O(n^2)$

⑤ $n + (n-1) + (n-2) + \cdots + 3 + 2 + 1$

$\Rightarrow (n+1)\frac{n}{2} \Rightarrow O(n^2)$



$\frac{n^2}{2} \Rightarrow O(n^2)$

⑥ $O(n)$

⑦ $O(\log n)$

⑨ $O(n \log n)$

⑩ $O(2^n)$

$1 + 2 + 4 + 8 \cdots 2^{n-1}$

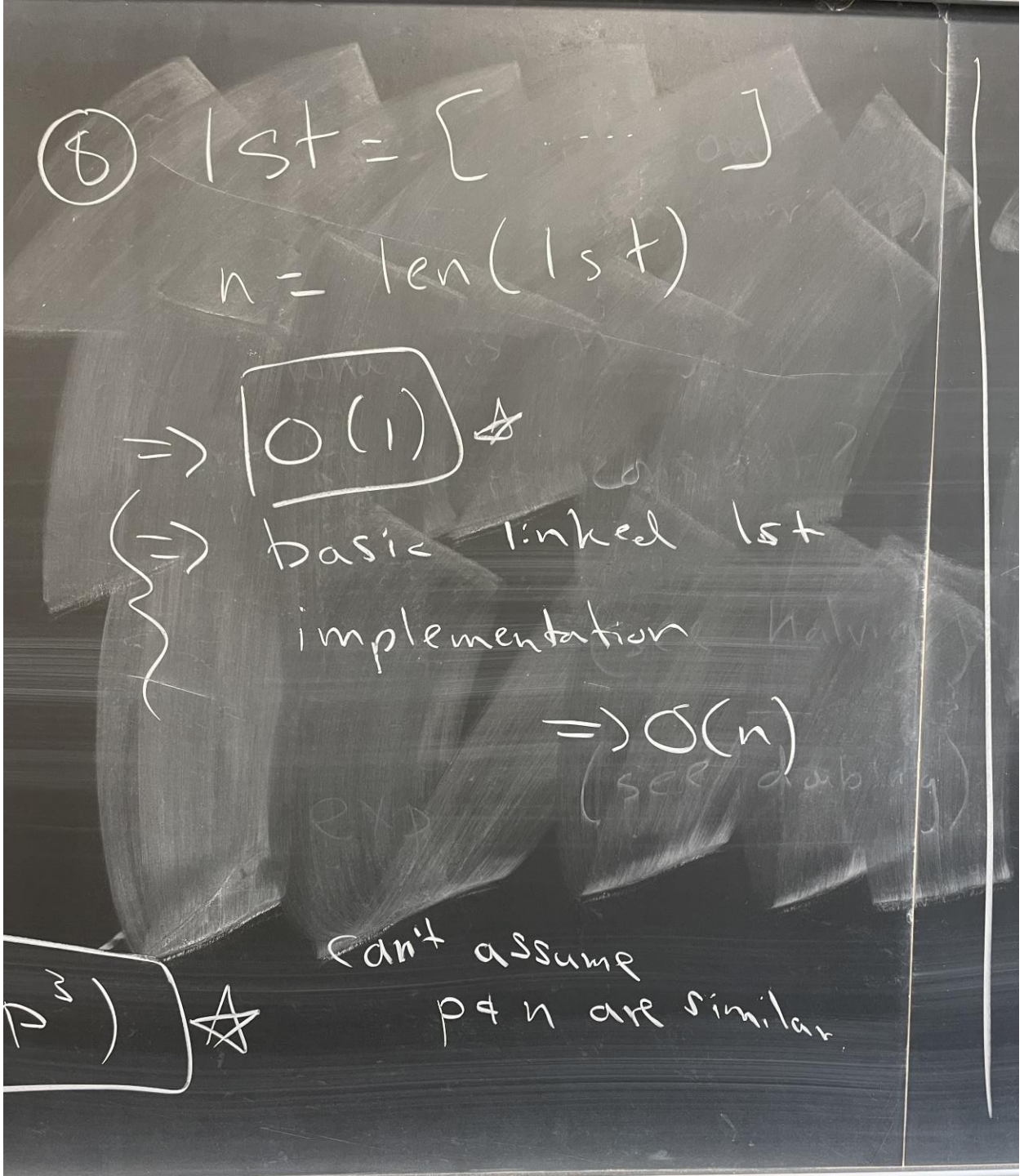
$1 + 2^1 + 2^2 + 2^3 + \cdots 2^{n-1} = S$

$(-)\quad 2^1 + 2^2 + 2^3 + \cdots 2^{n-1} + 2^n = 2S$

$1 - 2^n = S - 2S$

$S = 2^n - 1$

$\Rightarrow O(2^n)$


⑧ lst = [...]
n = len(lst)
⇒ O(1) ☆
⇒ basic linked lst
implementation
⇒ O(n)
can't assume
p & n are similar.
p³) ☆

(11) $\vec{w} = \underbrace{\underbrace{(X^T X)^{-1}}_{(a)\ (b)} \underbrace{X^T}_{(c)}}_{(d)} \vec{y}$

$(p+1)\times(p+1)$ & $(p+1)\times n$

(0) $X^T \Rightarrow O(np)$

(a) multiply $(p+1)\times n$ & $n\times(p+1)$ $\Rightarrow$ $O(p^2 n)$

(b) invert $(p+1)\times(p+1)$ matrix $\Rightarrow$ $O(p^3)$

inverse is cubic

(c) multiply $(p+1)\times n$ & $(n\times 1)$ $\Rightarrow$ $O(pn)$

(d) multiply $(p+1)\times(p+1)$ & $(p+1)\times 1$ $\Rightarrow$ $O(p^2)$

$\Rightarrow O(p^2 n + p^3 + pn + p^2) \Rightarrow \boxed{O(p^2 n + p^3)}$ ☆

(8) 1st

$n =$

$\Rightarrow$ O(

(=> bas

im

ea

matrix multiplication

$A \Rightarrow (n, p)$

$B \Rightarrow (p, m)$

$\Rightarrow O(nmp)$

$C \Rightarrow (p, p)$

$D \Rightarrow (p, m)$

$\Rightarrow O(p^2 m)$

inverse: matrix must be square

$\Rightarrow (n \times n)$

$\Rightarrow \boxed{O(n^3)}$

Outline

- Review runtime
- Handout 12
- Finish algorithmic bias
- Discussion: admissions at Haverford

How can we tell if an algorithm is biased?

D: dataset with attributes X, Y

* X is protected
* Y is unprotected (other features)

How can we tell if an algorithm is biased?

D: dataset with attributes X, Y

* X is protected
* Y is unprotected (other features)

Goal: determine outcome C (hired, admitted, etc)

How can we tell if an algorithm is biased?

D: dataset with attributes X, Y

* X is protected
* Y is unprotected (other features)

Goal: determine outcome C (hired, admitted, etc)

Direct discrimination: C = f(X)

* Female instrumentalist not hired for orchestra
* Some ethnic groups not allowed to eat at a restaurant

"Certifying and Removing Disparate Impact" Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, Suresh Venkatasubramanian

How can we tell if an algorithm is biased?

D: dataset with attributes X, Y

* X is protected
* Y is unprotected (other features)

Goal: determine outcome C (hired, admitted, etc)

Indirect discrimination: C = f(Y)

* but strong correlation between X and Y
* Ex: housing loans
* Ex: programming experience

"Certifying and Removing Disparate Impact" Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, Suresh Venkatasubramanian

Idea: if we can predict X from Y, could be **disparate impact** →

$$P(C=1 \mid X=0) \leq 0.8\, P(C=1 \mid X=1)$$

C: hired, admitted

$C=0$ not hired

Predictor $f: Y \to X$ (naive bayes)

Y: input, X: output

Metric

Balanced error rate (BER)

$$\varepsilon = BER = \frac{1}{2}\left[P(f(Y)=0 \mid X=1) + P(f(Y)=1 \mid X=0) \right]$$

Want high! i.e. confusion.

$P(f(Y)=0 \mid X=1) = \frac{10}{1000}$

$P(f(Y)=1 \mid X=0) = \frac{2}{50}$

① train a classifier $f(Y) \Rightarrow X$
(predict X)

② if BER is too low,
could be disparate impact

normal error $\Rightarrow \frac{10+25}{1000+50} \Rightarrow$ low

$$BER = \left(\frac{10}{1000} + \frac{25}{50}\right)\frac{1}{2}$$

$$= \text{high}$$

Outline

- Review runtime
- Handout 12
- Finish algorithmic bias
- Discussion: admissions at Haverford

Discussion: admissions at Haverford

- Haverford has suddenly started receiving 10x more applications than usual
- You are tasked with creating an algorithm to determine whether or not an applicant should be admitted
- Questions:
  - How would you encode features?
  - How would you use past admission data to train?
  - What cost function are you trying to optimize?