# COMPSCI 614: Randomized Algorithms with Applications to Data Science

Prof. Cameron Musco

University of Massachusetts Amherst. Spring 2024. Lecture 13 (Midterm Review)

- Thursday in class 10-11:15 am.
- Closed book, no calculator.
- Study guide posted under course schedule on the midterm line.
- Practice midterm posted in Moodle.

# Midterm Format/Content

#### Rough Format:

- 1. Set of true/false or always/sometimes/never questions
- 2. Set of short answer questions.
- 3-4. Two multi-part 'pr<u>oblem se</u>t like' questions.
- 5. One bonus question.

# Midterm Format/Content

#### Rough Format:

- 1. Set of true/false or always/sometimes/never questions
- 2. Set of short answer questions.
- 3-4. Two multi-part 'problem set like' questions.
- 5. One bonus question.

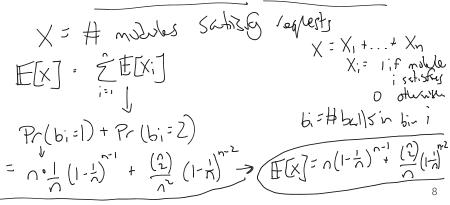
#### Content:

- See study guide for a detailed list.
- Should know key tools/analysis approaches. But don't need to memorize derivations from class.
- It will help to be familiar with homework problems, but don't need to memorize answers/derivations from them.

- I would focus on doing practice problems more than reviewing material.
- Definitely do the practice exam in semi-timed, closed note environment.
- Try to do as many other practice questions from the study guide, by redoing homework questions/problems given in class, etc.

A parallel computer consists of n processors and n memory modules. During a step, each processor sends a memory request to a random module. A module that receives 1 or 2 requests satisfies its requests; modules that receive more than two requests will not satisfy them. What is the expected number of modules that satisfy their requests?

A parallel computer consists of n processors and n memory modules. During a step, each processor sends a memory request to a random module. A module that receives 1 or 2 requests satisfies its requests; modules that receive more than two requests will not satisfy them. What is the expected number of modules that satisfy their requests?

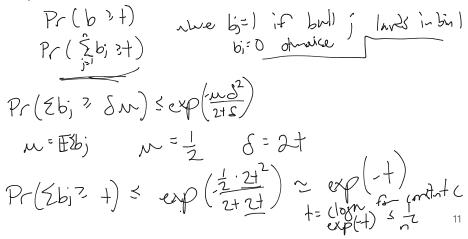


You store n items in a hash table with 2n buckets using a fully frandom hash function. Give an upper bound on the maximum load in any bucket, which holds with probability at least 1 - 1/n. Does the answer change significantly if you have n instead of 2n buckets?

$$F[X] = n(l + \frac{n}{2})^{n-1} + \frac{\binom{n}{2}}{\binom{n}{2}}(l + \frac{1}{2})^{n}$$

$$\approx \frac{n}{e} + \frac{n(n-1)}{2n} \cdot \frac{1}{e} = n(\frac{1}{e} + \frac{1}{2e})$$

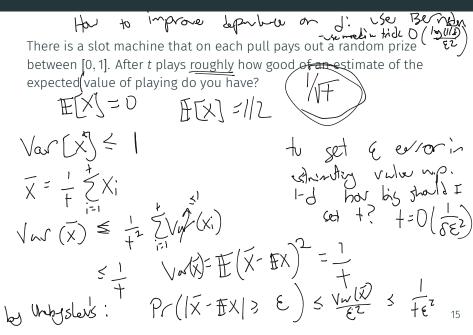
You store n items in a hash table with 2n buckets using a fully random hash function. Give an upper bound on the maximum load in any bucket, which holds with probability at leas 1 - 1/n. Does the answer change significantly if you have *n* instead of 2*n* buckets?  $\begin{array}{rcl} mh^{\chi} & hon^{\chi} & \sim & O\left(\frac{1}{9}m\right) \\ mx^{\chi} & \log d & \leq & O(\log n) \end{array}$  $Pr(mix bal 2t) = Pr(b_1 2t \text{ or } b_2 2t ... \text{ or } b_n 2t)$ (bi 2+) we need to set + st  $\left(\frac{Pr(b; 2+)}{n} \le \frac{1}{n^2}\right)$ Two  $Pr(max \log 2 \le +) \ge 1 - nPr(b; 2+)$ 31-1-31-1Consider the scenario above, where you use linear probing instead of chaining. Does the expected look up time change significantly if you use *n* rather than 2*n* buckets?



You have an algorithm that succeeds with probability 2/3. You run it t times independently and would like to ensure that the probability that the algorithms fails on a majority (i.e., > t/2) of these trials is at most  $\delta$ . How large should you set t?

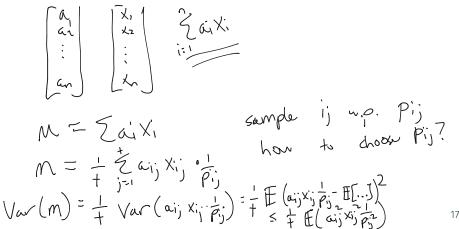
You have an algorithm that succeeds with probability 2/3. You run it t times independently and would like to ensure that the probability that the algorithms fails on a majority (i.e., > t/2) of these trials is at most  $\delta$ . How large should you set t?

Y= Hrailwes E[4]= +3 You have an algorithm that succeeds with probability 2/3. You run it t times independently and would like to ensure that the probability that the algorithm fails on a majority (i.e., > t/2) of these trials is at most  $\delta$ . How large should you set t?  $P_{\gamma}(|X - EX| \ge \frac{1}{2})$  $\leq$ notion to can boosted SUCLESS +=O(IO(IG))+=0(1/2) Nolsion  $Pr(Y \ge 1) = Pr(Y \ge 1) = 2$ 14



There is a slot machine that on each pull pays out a random prize between [0, 1]. After *t* plays roughly how good of an estimate of the expected value of playing do you have?

You would like to estimate the inner product of a vector  $a \in \mathbb{R}^n$  with a vector  $x \in \mathbb{R}^n$ . You know a completely but don't know x. Describe how to obtain an estimate of the inner product without reading all of x via important sampling.



You would like to estimate the inner product of a vector  $a \in \mathbb{R}^n$  with a vector  $x \in \mathbb{R}^n$ . You know a completely but don't know x. Describe how to obtain an estimate of the inner product without reading all of x via important sampling.

$$V_{av}(m) \leq \frac{1}{t} \left[ F(a_{ij}^{v}, X_{ij}^{v}, p_{ij}^{v}) \right]$$

$$= \int_{t}^{2} P_{j} \cdot a_{j}^{v} X_{ij}^{v} p_{j}^{v}$$

$$= \int_{t}^{2} \frac{a_{ij}^{v} X_{ij}^{v}}{P_{j}^{v}} p_{j}^{v} + \frac{1}{p_{j}^{v}} p_{j}^{v} + \frac{1}{p_{j}^{$$

 $\|a\|_{\infty}$ ,  $\|X\|_{\infty} \leq M$  $mn \sqrt{-1} \frac{2}{r} \frac{\alpha_j^2 x_j^2}{p_j} = \frac{1}{r} \frac{2}{r} \alpha_j^2 x_j^2$   $\frac{\sqrt{V_{+}}}{r} = \frac{1}{r} \frac{2}{r} \alpha_j x_j^2$   $\frac{\sqrt{V_{+}}}{r}$ sit pie [0,1] <u>76</u> = <u>76</u>  $2p_{j}=1$ set pit QailXj 50 m sample  $\sim$ 19

