Final In-Class Exam

(1:05 PM – 2:20 PM : 75 Minutes)

- This exam will account for either 15% or 30% of your overall grade depending on your relative performance in the midterm and the final. The higher of the two scores (midterm and final) will be worth 30% of your grade, and the lower one 15%.
- There are three (3) questions, worth 75 points in total. Please answer all of them in the spaces provided.
- There are 12 pages including four (4) blank pages and one (1) page of appendix. Please use the blank pages if you need additional space for your answers.
- The exam is *open slides* and *open notes*.

Good Luck!

Question	Pages	Score	Maximum
1. Escaping with a PhD	2–3		30
2. Finding Max	5–7		35
3. Files on Compact Discs	9		10
Total			75

Name: _____

Question 1. [30 Points] Escaping with a PhD. Once n PhD students got trapped indefinitely in a research lab, and they formed m teams $T_1, T_2, \ldots, T_m$ to work out various aspects of a successful escape strategy (i.e., getting a PhD). A PhD student could be part of any number of teams, and each team could include any number of PhD students. Together they were able to figure out how to get approximately half of the students out of the lab by negotiating approximately $\frac{n}{2}$ PhD degrees with the lab director. But the teams were so effective in finding this solution that the students that were going to be left behind would want to use them again to find another successful escape strategy. So the students decided to choose approximately half of the n students in such a way that even when that half leave the lab each team T_i would still have roughly $|T_i|/2$ of its original members in the lab.

For $1 \leq i \leq m$, let n_i be the number of members of team T_i before the espace, and let n_i' be the number after the escape. Let $d_i = \left|n_i' - \frac{n_i}{2}\right|$.

The students would like to choose the escaping party in such a way that $D = \max_{1 \leq i \leq m} d_i$ is minimized.

They came up with a surprisingly simple algorithm! Each of the n students was chosen to be included in the escaping party (i.e., chosen for a PhD) independently at random with probability $\frac{1}{2}$. You are asked to show that this is a reasonably good algorithm.

1(a) [**20 Points**] Show that if the simple algorithm is used then for each $i \in [1, m]$, $d_i < \sqrt{3 n_i \log m}$ w.h.p. in m.

Hint: $\Pr(|X - \mu| \geq \gamma) = \Pr(X - \mu \geq \gamma) + \Pr(\mu - X \geq \gamma)$.

1(b) [**10 Points**] Use your result from part 2(a) to show that $D < \sqrt{3n \log m}$ w.h.p. in m for the algorithm devised by the students.

Hint: Use the union bound, i.e., $\Pr\left(\cup_{1 \leq i \leq m} E_i\right) \leq \sum_{1 \leq i \leq m} \Pr\left(E_i\right)$. Also observe that $n_i \leq n$.

Use this page if you need additional space for your answers.

Question 2. [35 Points] Finding Max. We saw in the class how to compute the maximum of n numbers in $\Theta(n)$ work and $\Theta\left(\log^2 n\right)$ span (e.g., use the parallel prefix sums algorithm given in slides 70–74 of lecture 13 with **max** as the binary associative operator $\oplus$). In this problem, we will design a parallel algorithm with a shorter span. We will assume that the span of a parallel ***for*** loop is $\mathcal{O}(\log m + t)$, where m is the number of iterations, and t is the maximum span of an iteration.

2(*a*) **[8 Points]** Design an algorithm to find the maximum number in $A[1:n]$ in $\Theta\left(n^2\right)$ work and $\Theta(\log n)$ span. We assume that if multiple processors try to write to the same memory location at the same time only one of them (an arbitrary one) succeeds, and all others immediately fail. Let's call this algorithm Find-Max-2a.

Hint: Compare all pairs of numbers in parallel. If all processors trying to write to the same memory location simultaneously try to write the same value v to that location then no matter which of those processors succeeds, the value v will always be written. You may find allowing such benign races useful.

2(*b*) [**20 Points**] Write recurrences for the work and span of FIND-MAX-2B given below, and solve them.

Hint: For finding $T_1(n)$ you may want to use the substitution $T_1(n) = nT'_1(n)$ and solve for $T'_1(n)$ first. For $T_\infty(n)$ you may want to substitute $T(2^k) = T'_\infty(k)$ and solve for $T'_\infty(k)$ first.

FIND-MAX-2B($A[\,1 : n\,]$)

(Input is an array $A[\,1 : n\,]$ of n numbers, where $n = 2^{2^h}$ for some integer $h \geq 0$. This function returns the value of the maximum number in the given array.)

1. ***if*** $n = 2$ ***then return*** $\max\{\ A[1],\ A[2]\ \}$ {*base case*}
2. ***else***
3. Let $A_1[1 : \sqrt{n}], \ldots, A_{\sqrt{n}}[1 : \sqrt{n}]$ be the $\sqrt{n}$ consecutive segments of $A[1 : n]$ of size $\sqrt{n}$ each
4. ***array*** $B[\,1 : \sqrt{n}\,]$ {*temporary storage*}
5. ***parallel for*** $i \leftarrow 1$ ***to*** $\sqrt{n}$ ***do*** {*use a constant depth* ***parallel for*** *loop*}
6. $B[i] \leftarrow$ FIND-MAX-2B($A_i[\,1 : \sqrt{n}\,]$) {*recursively find the max in the i-th segment*}
7. ***return*** FIND-MAX-2A($B[\,1 : \sqrt{n}\,]$) {*use the algorithm from part* 2(*a*)}

Figure 1: Find the maximum number in $A[\,1 : n\,]$.

2(c) [**7 Points**] Is the algorithm in part 2(b) work-optimal? Why or why not?

Use this page if you need additional space for your answers.

Question 3. [**10 Points**] **Files on Compact Discs.** I have $m > 0$ files and a set S of $n > 1$ compact discs (CDs). I have copied each file to at least one of the CDs in S. Different files may have been copied to different sets of CDs. Now given that for each file I know which CDs I copied them to, I want to find a subset $S' \subseteq S$ such that each file is contained in at least one CD of S', and $|S'|$ is as small as possible.

3(a) [**10 Points**] Give a polynomial-time approximation algorithm for solving this problem. What is the approximation ratio?

Hint: *Map this problem to one of the problems you saw in the class, and use the approximation algorithm for that problem to solve the problem in this task.*

Use this page if you need additional space for your answers.

Use this page if you need additional space for your answers.

Appendix I: Useful Tail Bounds

Markov's Inequality. Let X be a random variable that assumes only nonnegative values. Then for all $\delta > 0$, $Pr[X \geq \delta] \leq \frac{E[X]}{\delta}$.

Chebyshev's Inequality. Let X be a random variable with a finite mean $E[X]$ and a finite variance $Var[X]$. Then for any $\delta > 0$, $Pr[|X - E[X]| \geq \delta] \leq \frac{Var[X]}{\delta^2}$.

Chernoff Bounds. Let $X_1, \ldots, X_n$ be independent Poisson trials, that is, each X_i is a 0-1 random variable with $Pr[X_i = 1] = p_i$ for some p_i. Let $X = \sum_{i=1}^{n} X_i$ and $\mu = E[X]$. Following bounds hold:

Lower Tail:

- for $0 < \delta < 1$, $Pr[X \leq (1-\delta)\mu] \leq \left(\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}}\right)^{\mu}$
- for $0 < \delta < 1$, $Pr[X \leq (1-\delta)\mu] \leq e^{-\frac{\mu\delta^2}{2}}$
- for $0 < \gamma < \mu$, $Pr[X \leq \mu - \gamma] \leq e^{-\frac{\gamma^2}{2\mu}}$

Upper Tail:

- for any $\delta > 0$, $Pr[X \geq (1+\delta)\mu] \leq \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mu}$
- for $0 < \delta < 1$, $Pr[X \geq (1+\delta)\mu] \leq e^{-\frac{\mu\delta^2}{3}}$
- for $0 < \gamma < \mu$, $Pr[X \geq \mu + \gamma] \leq e^{-\frac{\gamma^2}{3\mu}}$

Appendix II: The Master Theorem

Let $a \geq 1$ and $b > 1$ be constants, let $f(n)$ be a function, and let $T(n)$ be defined on the nonnegative integers by the recurrence

$$T(n) = \begin{cases} \Theta(1), & \text{if } n \leq 1, \\ aT\left(\frac{n}{b}\right) + f(n), & \text{otherwise,} \end{cases}$$

where, $\frac{n}{b}$ is interpreted to mean either $\lfloor \frac{n}{b} \rfloor$ or $\lceil \frac{n}{b} \rceil$. Then $T(n)$ has the following bounds:

Case 1: If $f(n) = \mathcal{O}\left(n^{\log_b a - \epsilon}\right)$ for some constant $\epsilon > 0$, then $T(n) = \Theta\left(n^{\log_b a}\right)$.

Case 2: If $f(n) = \Theta\left(n^{\log_b a} \log^k n\right)$ for some constant $k \geq 0$, then $T(n) = \Theta\left(n^{\log_b a} \log^{k+1} n\right)$.

Case 3: If $f(n) = \Omega\left(n^{\log_b a + \epsilon}\right)$ for some constant $\epsilon > 0$, and $af\left(\frac{n}{b}\right) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large n, then $T(n) = \Theta(f(n))$.