In-Class Final Exam

(11:35 AM – 12:50 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 four (4) questions, worth 75 points in total. Please answer all of them in the spaces provided.
- There are 14 pages including four (4) blank pages. Please use the blank pages if you need additional space for your answers.
- The exam is *open slides*. So you can consult the lecture slides during the exam. No additional cheatsheets are allowed.

Good Luck!

Question	Score	Maximum
1. Binary Addition		25
2. Nuts and Bolts		15
3. Partial Sums		10
4. Sampling		25
Total		75

Name: _____

Question 1. [25 Points] Binary Addition. Function Standard-Binary-Addition is the standard grade school algorithm for adding two n-bit binary numbers, and Rec-Binary-Addition is a recursive divide-and-conquer algorithm for the same task. In this problem we will try to parallelize the two algorithms and analyze the resulting parallel algorithms.

Standard-Binary-Addition($x_n \dots x_1,\ y_n \dots y_1$)

(Inputs are two n-bit binary numbers $X = x_n \dots x_1$ and $Y = y_n \dots y_1$, where $n \geq 1$. Output is an $(n+1)$-bit binary number $Z = z_{n+1} \dots z_1$, where Z is the sum of X and Y (i.e., $Z = X + Y$).)

1. $c \leftarrow 0$
2. ***for*** $i \leftarrow 1$ ***to*** n ***do***
3. $\quad s \leftarrow x_i + y_i + c$
4. $\quad z_i \leftarrow s$ ***mod*** $2,\ \ c \leftarrow s$ ***div*** 2
5. $z_{n+1} \leftarrow c$
6. ***return*** Z

Rec-Binary-Addition($x_n \dots x_1,\ y_n \dots y_1$)

(Inputs are two n-bit binary numbers $X = x_n \dots x_1$ and $Y = y_n \dots y_1$, where $n \geq 1$ is assumed to be a power of 2. Outputs are two $(n+1)$-bit binary numbers $Z^c = z^c_{n+1} \dots z^c_1$ and $Z = z_{n+1} \dots z_1$, where Z^c is the sum of X and Y assuming an initial carry (i.e., $Z^c = X + Y + 1$), and Z is the same sum without an initial carry (i.e., $Z = X + Y$).)

1. ***if*** $n = 1$ ***then***
2. $\quad$ ***if*** $x_1 = y_1 = 0$ ***then return*** $\langle\ 01,\ 00\ \rangle$
3. $\quad$ ***elif*** $x_1 = y_1 = 1$ ***then return*** $\langle\ 11,\ 10\ \rangle$
4. $\quad$ ***else return*** $\langle\ 10,\ 01\ \rangle$
5. ***else***
6. $\quad$ ***let*** $X_h = x_n \dots x_{\frac{n}{2}+1}$ ***and*** $X_l = x_{\frac{n}{2}} \dots x_1$ $\quad$ {*split X at the midpoint*}
7. $\quad$ ***let*** $Y_h = y_n \dots y_{\frac{n}{2}+1}$ ***and*** $Y_l = y_{\frac{n}{2}} \dots y_1$ $\quad$ {*split Y at the midpoint*}
8. $\quad \langle\ L^c,\ L\ \rangle \leftarrow$ Rec-Binary-Addition$(\ X_l,\ Y_l\ )$ $\quad \left\{ \text{where } L^c = l^c_{\frac{n}{2}+1} l^c_{\frac{n}{2}} \dots l^c_1 \text{ and } L = l_{\frac{n}{2}+1} l_{\frac{n}{2}} \dots l_1 \right\}$
9. $\quad \langle\ H^c,\ H\ \rangle \leftarrow$ Rec-Binary-Addition$(\ X_h,\ Y_h\ )$ $\quad \left\{ \text{where } H^c = h^c_{\frac{n}{2}+1} h^c_{\frac{n}{2}} \dots h^c_1 \text{ and } H = h_{\frac{n}{2}+1} h_{\frac{n}{2}} \dots h_1 \right\}$
10. $\quad$ ***if*** $l_{\frac{n}{2}+1} = 1$ ***then*** Copy$\Big(\ Z,\ H^c,\ l_{\frac{n}{2}} \dots l_1\ \Big)$ $\quad \left\{ (\text{carry from position } \tfrac{n}{2}) \Rightarrow Z = h^c_{\frac{n}{2}+1} h^c_{\frac{n}{2}} \dots h^c_1 l_{\frac{n}{2}} \dots l_1 \right\}$
11. $\quad$ ***else*** Copy$\Big(\ Z,\ H,\ l_{\frac{n}{2}} \dots l_1\ \Big)$ $\quad \left\{ (\text{no carry from position } \tfrac{n}{2}) \Rightarrow Z = h_{\frac{n}{2}+1} h_{\frac{n}{2}} \dots h_1 l_{\frac{n}{2}} \dots l_1 \right\}$
12. $\quad$ ***if*** $l^c_{\frac{n}{2}+1} = 1$ ***then*** Copy$\Big(\ Z^c,\ H^c,\ l^c_{\frac{n}{2}} \dots l^c_1\ \Big)$ $\quad \left\{ (\text{carry from position } \tfrac{n}{2}) \Rightarrow Z^c = h^c_{\frac{n}{2}+1} h^c_{\frac{n}{2}} \dots h^c_1 l^c_{\frac{n}{2}} \dots l^c_1 \right\}$
13. $\quad$ ***else*** Copy$\Big(\ Z^c,\ H,\ l^c_{\frac{n}{2}} \dots l^c_1\ \Big)$ $\quad \left\{ (\text{no carry from position } \tfrac{n}{2}) \Rightarrow Z^c = h_{\frac{n}{2}+1} h_{\frac{n}{2}} \dots h_1 l^c_{\frac{n}{2}} \dots l^c_1 \right\}$
14. $\quad$ ***return*** $\langle\ Z^c,\ Z\ \rangle$

Copy($z_{2n+1} \dots z_1,\ h_{n+1} \dots h_1,\ l_n \dots l_1$)

1. ***for*** $i \leftarrow 1$ ***to*** n ***do***
2. $\quad z_i \leftarrow l_i$
3. $\quad z_{n+i} \leftarrow h_i$
4. $z_{2n+1} \leftarrow h_{n+1}$

1(a) [**3 Points**] Can you replace the ***for*** loop in lines 2–4 of STANDARD-BINARY-ADDITION with a ***parallel for*** loop (without changing anything else)? Justify your answer.

1(b) [**6 Points**] Which parts of REC-BINARY-ADDITION can be executed in parallel? Justify your answer.

1(c) [**10 Points**] Write down the recurrence relations for *work* and *span* for your parallel version of Rec-Binary-Addition from part 1(b), and solve them. Assume that the span of a ***parallel for*** loop with n iterations is $\Theta(\log n) + k$, where k is the maximum span of one iteration.

1(d) [**4 Points**] Find the parallel running time (i.e., T_p) and the parallelism of the parallel Rec-Binary-Addition from part 1(b).

1(e) [**2 Points**] Is your parallel Rec-Binary-Addition work-optimal? Justify your answer.

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

Question 2. [**15 Points**] **Nuts and Bolts.** Suppose you are given n nuts and n bolts. Each nut has unique size and it matches exactly one bolt and vice versa. Your job is to find the matching bolt for every nut in the set. When you solve this problem you can only compare a nut with a bolt and conclude that either the nut exactly fits the bolt or it is too small for the bolt or it is too large. You are not allowed to compare one nut with another nut or a bolt with another bolt.

2(a) [**5 Points**] Suppose you have chosen a nut uniformly at random from the given set. Show that in $\Theta(n)$ time you can find the matching bolt, and divide the remaining nuts and bolts into two sets: one containing all nuts and bolts smaller than the chosen pair and the other containing everything larger.

2(b) [**10 Points**] Prove that using the partition algorithm in part 2(a) you can solve the nuts and bolts problem in $\mathcal{O}(n \log n)$ time with high probability. You do not necessarily need to prove the bound explicitly. You can simply argue that your algorithm is structurally exactly similar to an algorithm you have already seen in the class, and so the bound proved for that algorithm applies.

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

Question 3. [**10 Points**] **Partial Sums.** In the class we have shown that the *partial semigroup sums* problem can be solved using $\Theta(n\alpha(n))$ space with $\mathcal{O}(\alpha(n))$ applications of the associative operator per query. How do you reduce the space complexity to $\Theta(n)$ without changing the query complexity asymptotically?

Question 4. [25 Points] Sampling. In this problem we will analyze the properties of two different sampling of elements from an array.

4(*a*) **[10 Points]** One can show that given an array A of n distinct numbers no deterministic algorithm can find an element among the smallest $\frac{n}{k}$ members of A using fewer than $\left(\frac{k-1}{k}\right) n$ comparisons. But consider the function given below that runs in $\Theta\left(\log_{\left(\frac{k}{k-1}\right)} n\right)$ time. Prove that with high probability in n the element it returns is among the smallest $\frac{n}{k}$ elements of A.

```
TOP( A[ 1 : n ], k )
(Inputs are an array A[ 1 : n ] of n distinct numbers, and an integer k ∈ [1, n]. Output is an element x of A.)
1. x ← A[ RANDOM( 1, n ) ]                    {choose a number from A uniformly at random}
2. for i ← 1 to ⌈log_(k/(k−1)) n⌉ do
3.     y ← A[ RANDOM( 1, n ) ]                {choose a number from A uniformly at random}
4.     if y < x then x ← y                    {keep the smaller number}
5. return x
```

4(*b*) **[15 Points]** Consider the function given below, and prove that with high probability in n no $\left\lceil \frac{2k}{k-1} \right\rceil$ numbers in $B[\ 1 : m\ ]$ are equal.

SAMPLE($A[\ 1 : n\ ],\ k$)
(Inputs are an array $A[\ 1 : n\ ]$ of n distinct numbers, and a real number $k > 1$. Output is an array $B[\ 1 : m\ ]$ of $m = \left\lfloor n^{\frac{1}{k}} \right\rfloor$ numbers sampled uniformly at random from A.)

1. ***array*** $B[\ 1 : m\ ]$
2. ***for*** $i \leftarrow 1$ ***to*** m ***do***
3. $\quad j \leftarrow$ RANDOM($1, n$) $\quad$ {*choose an integer uniformly at random from* $[\ 1, n\ ]$}
4. $\quad B[\ i\ ] \leftarrow A[\ j\ ]$
5. ***return*** B

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

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