Verify: There are 15 pages in this examination, comprising 6 questions worth a total of 100 points. The last 2 pages are an appendix with costs of sequence, set and table operations.

Write: the name (e.g., “J. Snow”) of the persons sitting to your left and to your right below your andrew id (in left to right order).

Time: You have 80 minutes to complete this examination.

Goes without saying: Please answer all questions in the space provided with the question. Clearly indicate your answers.

Beware: You may refer to your one double-sided $8\frac{1}{2} \times 11$in sheet of paper with notes, but to no other person or source, during the examination.

Primitives: In your algorithms you can use any of the primitives that we have covered in the lecture, unless otherwise states. A reasonably comprehensive list is provided at the end and sometimes in the body of the question itself.

Code: When writing your algorithms, you can use ML syntax or the pseudocode notation used in the notes or in class. In the questions, we use pseudocode.

Good luck!
<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>True/False</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>Recurrences</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>Short Answer Problems</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>Exclamations</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>Iterate and Reduce</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>Random Other Questions</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
Question 1: True/False  (16 points)
Please Circle your choice.

(a) (2 points) **TRUE** or **FALSE**: If you can reduce comparison-based sorting to your problem in \( O(n) \) work, then you can solve your problem in \( \Theta(n \log n) \) work.

(b) (2 points) **TRUE** or **FALSE**: Since we can reduce the shortest superstring (SS) problem to the Traveling Salesperson Problem (TSP) using polynomial work, the SS problem can be solved in polynomial work.

(c) (2 points) **TRUE** or **FALSE**: Parallelism is proportional to Work divided by Span.

(d) (2 points) **TRUE** or **FALSE**: The union bound implies that if \( n \) people each have probability \( 1/n^2 \) of having a twin sister, then the probability that any has a twin sister is exactly \( 1/n \).

(e) (2 points) **TRUE** or **FALSE**: The expressions \( \text{Seq.scan} \ f \ I \ A \) always takes \( O(\log |A|) \) span.

(f) (2 points) **TRUE** or **FALSE**: \( f(n) = \Theta(g(n)) \) implies \( f(n) = O(g(n)) \).

(g) (2 points) **TRUE** or **FALSE**: \( f(n) = O(g(n)) \) implies \( g(n) = \Omega(f(n)) \).

(h) (2 points) **TRUE** or **FALSE**: For independent random variables \( X \) and \( Y \):
\[
\]
Question 2: Recurrences  (12 points)

Give a closed-form solution in terms of $\Theta$ for the following recurrences. Also, state whether the recurrence is root dominated, leaf dominated, or approximately balanced in the recurrence tree (as defined in class). You do not have to show your work, but it might help you get partial credit.

(a) (2 points) $f(n) = f(n/2) + \log n$

(b) (2 points) $f(n) = 2f(n/2) + n^{1.25}$

(c) (2 points) $f(n) = 2f(n/2) + n \log n$

(d) (2 points) $f(n) = 3f(n/2) + n^{1/2}$

(e) (2 points) $f(n) = f(\sqrt{n}) + \log n$

(f) (2 points) $f(n) = \sqrt{n}f(\sqrt{n}) + n$
Question 3: Short Answer Problems (21 points)

(a) (7 points) Scan and reduce require associative functions and an identity. For each of the following binary functions circle it if is associative, and if it is associative, specify what the identity is.

- \( a \oplus b = a + b \)
- \( a \oplus b = a/b \)
- \( a \oplus b = a \times b \)
- \( a \oplus b = a \cup b \)
- \( (a, b) \oplus (c, d) = (\max(a, c), \min(b, d)) \)
- \( a \oplus b = a + \max(a, b) \)
- \( a \oplus b = \text{case } b \text{ of } \text{NONE } => a \ | \_ => b \)

(b) (6 points) Write a function to compute the factorial of all numbers from 1 to \( n \). The function should return a sequence of length \( n \), where the index \( i \) (starting at zero) stores the factorial of the number \( i + 1 \). Your solution should take \( O(n) \) work and \( O(\log n) \) span. You may assume multiplication is a unit cost operation. You may not simply use the \(!\) operator in your solution. (Our solution is one line.)

\[
\text{factorial (n : int) : int seq } =
\]

15–210 Exam I 5 of 15 24 February 2017
(c) (8 points) What is the asymptotic work and span of the following code for finding primes:

\[
\text{primes}(n) = \begin{align*}
\text{let } & \text{sieves} = \{ (i \times j, \text{false}) : 2 \leq i \leq \lceil \sqrt{n} \rceil, 2 \leq j < \lceil n/i \rceil \} \\
\text{in } & R = \text{inject(sieves, \{true : 0 \leq i \leq n\})} \\
\text{end} \\
\{ i : 2 \leq i \leq n | R[i] \}
\end{align*}
\]

\[
W(n) = \quad S(n) = \n\]
Question 4: Exclamations  (14 points)
Doctor Tooten is tired of seeing long lists of exclamation marks in a row!!!!!!! She therefore writes a linear work \(O(n)\) logarithmic span \(O(\log n)\) algorithm based on reduce that takes a sequence of \(n\) characters and reports back the length of the longest contiguous sequence of exclamation marks.

(a) (10 points) Please fill in the code for her algorithm below.

\[
\text{exclamation}(S) = \text{let}\\
\begin{align*}
(*) \text{ All}(n) & \text{ indicates all } n \text{ characters are !} \\
& \text{ Some}(s, e, m) & \text{ indicate the first } s \text{ characters are !, the last } e \text{ characters are !, and the longest contiguous sequence of ! not reaching the start or end has length } m 
\end{align*}
\]

\[
\text{datatype ex} = \text{ All of int} \\
| \text{ Some of } (\text{int} \times \text{int} \times \text{int})
\]

\[
\text{singleton}(v) = \text{ if } (x = '!'') \text{ then } \text{All}(1) \\
\text{else } \text{Some}(0,0,0)
\]

\[
I = \text{______________________________}
\]

\[
\text{combine}(a_1,a_2) = \\
\text{case } (a_1,a_2) \text{ of} \\
\begin{align*}
(\text{All}(n_1), \text{All}(n_2)) & \Rightarrow \text{______________________________} \\
| (\text{All}(n_1), \text{Some}(s_2,e_2,m_2)) & \Rightarrow \text{______________________________} \\
| (\text{Some}(s_1,e_1,m_1), \text{All}(n_2)) & \Rightarrow \text{______________________________} \\
| (\text{Some}(s_1,e_1,m_1), \text{Some}(s_2,e_2,m_2)) & \Rightarrow \text{______________________________}
\end{align*}
\]

\[
\text{in case } \text{reduce combine I } \langle \text{singleton}(x) : x \in S \rangle \text{ of} \\
\text{All}(n) \Rightarrow \text{______________________________} \\
| \text{Some}(s,e,m) \Rightarrow \text{______________________________}
\]

end

To get the following right, you must get (a) right.

(b) (2 points) **TRUE** or **FALSE**: Your function \(\text{combine}\) is associative.

(c) (2 points) **TRUE** or **FALSE**: If you replace \text{reduce} with \text{iterate} it will return the same result and in the same asymptotic work.
Question 5: Iterate and Reduce  (12 points)

Let's say we had an implementation of sequences such that \( \text{Seq.append}(A,B) \) takes \( \Theta(\sqrt{|A| + |B|}) \) work and \( \Theta(1) \) span. All other costs are the same as for array sequences. Please determine \( \Theta \) bounds for the work and span of the following functions in terms of \( n = |S| \).

(a) (6 points)

\[
\text{Seq.iterate Seq.append (Seq.empty()) (Seq.map Seq.singleton S)}
\]

\[
W(n) = \\
S(n) =
\]

(b) (6 points)

\[
\text{Seq.reduce Seq.append (Seq.empty()) (Seq.map Seq.singleton S)}
\]

\[
W(n) = \\
S(n) =
\]
Question 6: Random Other Questions  (25 points)

(a) (5 points) Consider a random sequence of $n$ bits. Briefly argue that there is at most a $1/n$ chance that this sequence will have a string of $2\log_2 n$ or more consecutive zeroes. You should include no more than two sentences.

For (b) and (c): You independently throw two unbiased 3-sided die (sides 1, 2, 3).

(b) (2 points) What is the expected sum?

(c) (2 points) What is the expected maximum value?

(d) (5 points) You have a stash of $k$ candy bars. Each day, when you get home, you roll a die. If it comes up 6, then you eat a candy bar, otherwise you don’t. In expectation, how many days will it take for you to eat all $k$ candy bars?

One more page.
(e) (5 points) $n$ people are standing in line to see a rock concert when all of a sudden the lead singer shows up at the ticket window and poses for a photo. You can only take a clear photo of her if you are taller than everyone in line in front of you. Furthermore everyone in line has a different height and they are standing in a random order. Write an expression for the exact expected number of people who can take a clear photo, and give what it approximately equals.

(f) (6 points) In a sequence $S$, we say that positions $i$ and $j$ have an inversion if $i < j$ but $S[i] > S[j]$. If elements in $S$ are all distinct and are in a random order, what is the expected total number of inversions? For instance, if $S$ was in reverse-sorted order, the total number of inversions would be $\binom{n}{2}$. 
Scratch Work:
Appendix: Library Functions

signature SEQUENCE =
  sig
    type 'a t
    type 'a seq = 'a t
    type 'a ord = 'a * 'a -> order
  datatype 'a listview = NIL | CONS of 'a * 'a seq
  datatype 'a treeview = EMPTY | ONE of 'a | PAIR of 'a seq * 'a seq
  exception Range
  exception Size

  val nth : 'a seq -> int -> 'a
  val length : 'a seq -> int
  val toList : 'a seq -> 'a list
  val toString : ('a -> string) -> 'a seq -> string
  val equal : ('a * 'a -> bool) -> 'a seq * 'a seq -> bool

  val empty : unit -> 'a seq
  val singleton : 'a -> 'a seq
  val tabulate : (int -> 'a) -> int -> 'a seq
  val fromList : 'a list -> 'a seq

  val rev : 'a seq -> 'a seq
  val append : 'a seq * 'a seq -> 'a seq
  val flatten : 'a seq seq -> 'a seq

  val filter : ('a -> bool) -> 'a seq -> 'a seq
  val map : ('a -> 'b) -> 'a seq -> 'b seq
  val zip : 'a seq * 'b seq -> ('a * 'b) seq
  val zipWith : ('a * 'b -> 'c) -> 'a seq * 'b seq -> 'c seq

  val enum : 'a seq -> (int * 'a) seq
  val filterIdx : (int * 'a -> bool) -> 'a seq -> 'a seq
  val mapIdx : (int * 'a -> 'b) -> 'a seq -> 'b seq
  val update : 'a seq * (int * 'a) -> 'a seq
  val inject : 'a seq * (int * 'a) seq -> 'a seq

  val subseq : 'a seq -> int * int -> 'a seq
  val take : 'a seq -> int -> 'a seq
  val drop : 'a seq -> int -> 'a seq
  val splitHead : 'a seq -> 'a listview
  val splitMid : 'a seq -> 'a treeview
val iterate : ('b * 'a -> 'b) -> 'b -> 'a seq -> 'b
val iteratePrefixes : ('b * 'a -> 'b) -> 'b -> 'a seq -> 'b seq * 'b
val iteratePrefixesIncl : ('b * 'a -> 'b) -> 'b -> 'a seq -> 'b seq
val reduce : ('a * 'a -> 'a) -> 'a -> 'a seq -> 'a
val scan : ('a * 'a -> 'a) -> 'a -> 'a seq -> 'a seq * 'a
val scanIncl : ('a * 'a -> 'a) -> 'a -> 'a seq -> 'a seq

val sort : 'a ord -> 'a seq -> 'a seq
val merge : 'a ord -> 'a seq * 'a seq -> 'a seq
val collect : 'a ord -> ('a * 'b) seq -> ('a * 'b seq) seq
val collate : 'a ord -> 'a seq ord
val argmax : 'a ord -> 'a seq -> int

val $ : 'a -> 'a seq
val % : 'a list -> 'a seq
end
<table>
<thead>
<tr>
<th>ArraySequence</th>
<th>Work</th>
<th>Span</th>
</tr>
</thead>
<tbody>
<tr>
<td>empty ()</td>
<td>O(1)</td>
<td>O(1)</td>
</tr>
<tr>
<td>singleton a</td>
<td></td>
<td></td>
</tr>
<tr>
<td>length s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nth s i</td>
<td></td>
<td></td>
</tr>
<tr>
<td>subseq s (i, len)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tabulate f n</td>
<td>O \left( \sum_{i=0}^{n-1} W_i \right)</td>
<td>O \left( \max_{i=0}^{n-1} S_i \right)</td>
</tr>
<tr>
<td>map f s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>zipWith f (s, t)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reduce f b s</td>
<td>O(n)</td>
<td>O(lg n)</td>
</tr>
<tr>
<td>scan f b s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>filter p s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>flatten s</td>
<td>O \left( \sum_{i=0}^{n-1} (1 +</td>
<td>s[i]</td>
</tr>
<tr>
<td>sort cmp s</td>
<td>O(n \lg n)</td>
<td>O(\lg^2 n)</td>
</tr>
<tr>
<td>merge cmp (s, t)</td>
<td>O(m + n)</td>
<td>O(\lg (m + n))</td>
</tr>
<tr>
<td>append (s, t)</td>
<td>O(m + n)</td>
<td>O(1)</td>
</tr>
<tr>
<td>inject (p, a)</td>
<td>O(m + n)</td>
<td>O(1)</td>
</tr>
</tbody>
</table>