Write your name and student id here: Peter Griffin

‘What makes a river so restful to people is that it doesn’t have any doubt - it is sure to get where it is going, and it doesn’t want to go anywhere else.’ Hal Boyle

Do not open this booklet until instructed.

- You are not allowed to use any printed/written material, laptops/cell-phones. Please turn off your cell phones and put them in your bags.
- Manage your time. We start the exam at 10:30 and end the exam at 11:25. You have 55 minutes.
- There are 6 pages (including this cover page). Write your answers in the provided boxes.
- In the unlikely case that you find the exam too long/hard, do not panic. The marks will be scaled so that the highest mark gets the full mark.
- There are more important things in life than this exam. So, relax and smile. Also, there are more important components to this course than the midterm. So, relax further and smile more (but still manage your time while relaxing).
1. Short Answer (20 marks)

Provide your short answers in the provided boxes. There is no need to justify your answers.

1. True or False: \( n^{1.001} \in \omega(n \log n) \)  
   True
   Note: In general, remember that \( n^\epsilon \) (for \( \epsilon > 0 \)) is asymptotically larger than poly-logarithmic functions like \( \log n \) or \( \log^{10000} n \).

2. True or False: \( \frac{n \log \log n}{\log n} \in o(n) \)  
   True
   Note: \( \log \log n \) is asymptotically smaller than \( \log n \).

3. Consider the following pseudocode:

   ```
   foo(n)
   1. i ← 1
   2. prod ← 1
   3. while i < min\{n, 2018\} do
   4.     for j = i to n do
   5.         prod ← prod \times j
   6.     i ← i \times 3
   7. return prod
   ```

   What is the worst-case running time of \( foo(n) \)?
   Express your answer using \( \Theta \)-notation in terms of \( n \), and be as precise as possible.

   \( \Theta(n) \)
   Note: In the asymptotic sense, \( \min\{n, 2018\} = 2018 \) (since we consider arbitrary large values of \( n \) in the asymptotic analysis). So, the while loop iterates a constant number of time. The second loop runs in \( \Theta(n) \) in a in its longest iteration.

4. Assume \( T(1) = 5 \) and \( T(n) = 9T(n/3) + n^2 \). Give an expression for the run-time of \( T(n) \) using \( \Theta \) notation. You might use Master theorem. Only the final answer is required.

   Case 2 of Master theorem gives \( T(n) \in \Theta(n^2 \log n) \)

5. True or false: The cost of QuickSort and QuickSelect are the same, if the same pivot-choosing algorithm is used.
   False
   Note: The cost of QuickSort is always more than QuickSelect. For example, with the best pivot-selection which takes the median, QuickSort runs in \( \Theta(n \log n) \) and QuickSelect runs in \( \Theta(n) \).

6. True or false: Quick-select runs in \( \Theta(n) \) in the average case.
   True

7. True or false: Using an augmented AVL tree, it is possible to find the median of \( n \) items in \( o(n) \).
   True
   Note: Median is a special case of selection, which can be done \( \Theta(\log n) \) in a properly augmented AVL tree.

8. True or false: A binary search tree can have height \( \Theta(n) \).
   True
   Note: consider a chain of nodes so that each internal node has only one node with smaller key on its left.

Marking Scheme: parts 3 and 4 each 3 marks; all others each 2 marks. You get the full mark or 0 for each part.
2. AVL Trees (10 marks)

Consider the following AVL tree $T$:

1. Write in the missing balance factors in the figure above.

2. Perform operation $\text{insert}(45)$ on $T$.
   Draw the tree before and after each rotation performed (no need to show balance factors).

Marking Scheme: Parts 1 has 3 marks; you lose 1 mark for each mistake in balancing factor. Part 2 has 7 marks: 2 marks which involves inserting 45 in the right place and applying the right rotation.
3. Skip Lists (10 marks)

Consider the skip-list $S$ shown below.

1. Show how $\text{Search}(S, 170)$ proceeds. More specifically show what nodes and in which order are visited. You should refer to the nodes using their keys and levels, e.g., you can say “node 104 at level 1”. The lowest level is level 0.

$(-\infty, 4) \rightarrow (-\infty, 3) \rightarrow (-\infty, 2) \rightarrow (104, 2) \rightarrow (104, 1) \rightarrow (104, 0) \rightarrow (169, 0) \rightarrow (179, 0)$.

Unsuccessful Search.

2. Show the skip list obtained by removing the key 287 from $S$, i.e., draw the skip list after performing $\text{Delete}(S, 287)$.

Marking Scheme:
Each part has 5 marks. You lose 2 marks in part 2 if you fail to remove the highest (top-most) level.
4. Binomial Heaps (10 marks)

Consider the following binomial heaps. Show the resulting heap when we apply the operation extract-max(). In case of merging trees, in case there were three binomial trees of the same order, merge the two ‘older’ trees (keep the new tree which is the product of previous merge). Show your work (intermediate steps).

Marking Scheme:  You lose 3 marks for each mistake in merging trees or maintaining the heap property.
5. Amortized Analysis (10 marks)

Consider a variant of dynamic arrays in which when an array becomes full, instead of doubling the size of the array, we quadruple the size of the array, i.e., we multiply it by four. This way, array sizes will be powers of 4. E.g., on the 65'th operation (insertion), the size of array is changed from 64 to 256. Use the potential-function method to find the amortized cost of each operation. You should use the potential function \( \Phi(i) = \frac{4}{3}i - \frac{1}{3}a_i \) where \( a_i \) is the size of the array.

For an inexpensive operation, the actual cost is just 1 which is for inserting an item into an array (which has enough space). The size of the array is not changed, i.e., \( a_i = a_{i-1} \) and the difference in potential is \( \Delta \Phi = (\frac{4}{3}i - \frac{1}{3}a_i) - (\frac{4}{3}(i - 1) - \frac{1}{3}a_{i-1}) = \frac{4}{3}(i - (i - 1)) = 4/3 \). So, the amortized cost is \( 1 + \Delta \Phi = 1 + 4/3 = 7/3 \).

For an expensive operation, the actual cost involves copying \( a_{i-1} \) item to the new array and additional unit for inserting the new item. So the actual cost is \( 1 + a_{i-1} \). The size of array is doubled, i.e., \( a_i = 4a_{i-1} \). So, the difference in potential is

\[
\Delta \Phi = \left(\frac{4}{3}i - \frac{1}{3}a_i\right) - \left(\frac{4}{3}(i - 1) - \frac{1}{3}a_{i-1}\right) \\
= \left(\frac{4}{3}i - \frac{1}{3}4(a_{i-1})\right) - \left(\frac{4}{3}(i - 1) - \frac{1}{3}a_{i-1}\right) = 4/3 - a_{i-1}. 
\]

So, the amortized cost is \( (1+a_{i-1}) + \Delta \Phi = (1+a_{i-1}) + (4/3 - a_{i-1}) = 7/3 \).

Marking Scheme: 5 marks for analysis of each of the two cases (expensive vs. inexpensive operations). In both cases, you should deduce the amortized cost of 7/3.