Al quiz rules from the course syllabus are in effect for the real quiz, in addition to what follows.

If you have questions about the test, please ask *on Piazza* as a **private** post, viewable only by you and the instructors. The instructors will make an announcement via a pinned errata post if something needs to be said to the entire class. **During the real quiz, do not make comments on any non-private posts on Piazza during the quiz for any reason. Making such comments may subject you to a grade penalty.**

This quiz is to be individual effort. Students are permitted to use notes, electronics, and bring textbooks. The work you submit for each quiz is expected to be produced by *you, alone and solely for this assessment.* You may not reuse or repurpose anything you wrote at another time, nor may you ever use anything written by someone else during this quiz. However, despite being allowed notes and electronics, you may *not* seek out the answer to a question in any way, nor may you communicate with anyone during the exam, *for any reason*, with the exception of asking a question on Piazza *set as instructors-only for visibility.*

You will have 40 minutes for this quiz, plus an additional ten minutes to enter your answers to the answer form. Please be careful when you enter your answers, as this will very likely be graded by a computer program and not a human. Students who have timed accommodations, such as through DSC, should use the 50 minute total as a baseline.

Future quizzes will also have free-response sections in addition to the constrained-answer style that are in quiz 1.

As this is the sample quiz, you may complete this any way you would like, and are not obligated to submit anything. However, I encourage you to allocate time as if this were the real quiz, as that will provide you some feedback about your preparation.

If you want to submit answers, feel free to use the following form to do so:

`https://docs.google.com/forms/d/e/1FAIpQLScJ8yVvKF7GAOawWW8G1ao4y-kU-55PLrnK9MQEMWqZmyaN9Q/viewform`

The real quiz will require submission at a similar form.

It is my intention to "grade" any submitted by Thursday morning and provide that feedback to such students. I will not start the grading prior to 8AM on Thursday April 8, so any submissions by then will be graded if I do so. Under no conditions will this "grading" count towards your actual grade in CompSci 161 : it is only for your feedback.

1. (1 point) Fór each of the following functions, determine whether it is $\mathcal{O}(n)$, $\Omega(n)$, or both. You do not need to provide proof or justification.
   Note that we are *not* asking you to provide the "best" $\mathcal{O}$-notation/Ω-notation; merely to describe it with one or both of the choices provided.

   $a(n) = n(\log n)^{10}$ ☐ $\mathcal{O}(n)$ ☐ $\Omega(n)$

   $b(n) = n^{0.99}$ ☐ $\mathcal{O}(n)$ ☐ $\Omega(n)$

   $c(n) = 10^{100}n$ ☐ $\mathcal{O}(n)$ ☐ $\Omega(n)$

2. (1.5 points) Rank the following functions in order from smallest asymptotic running time to largest. Additionally, identify a pair of functions x, y where $x(n) = \Theta(y(n))$. You do not need to show your work.

   For the ordering requirement, write the function identifier letters in order. Your submission on the "short answer" form should be exactly seven characters. For example, if you believe the functions are currently listed in asymptotic order, your answer should be "abcdefg" (without the quotes).

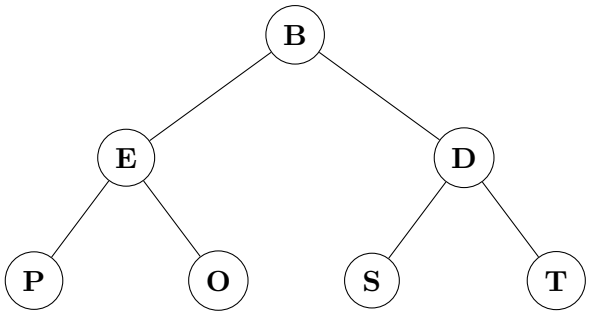
   For the Θ requirement, there are a series of checkboxes; check the ones that are Θ of one another. For example, if you believe $2^{n-2} - n$ is $\Theta(10^4!)$, check the boxes for $a(n)$ and $b(n)$ (and no others).

   (a) $a(n) = 2^{n-2} - n$
   (b) $b(n) = 10^4!$
   (c) $c(n) = n\log(64n) + \log n$
   (d) $d(n) = 8^{(\log n)+1}$
   (e) $e(n) = n^8 \log n$
   (f) $f(n) = n\log^8 n$
   (g) $g(n) = 2^{6\log\sqrt{n}}$

3. (1.0 point) In lecture, we saw that an array can be considered a complete tree and, if the heap property applies, it is also a heap. In previous classes, you saw that a string (data type) is an array of characters. If we use alphabetical ordering, the string "BEDPOST" is a min-heap.

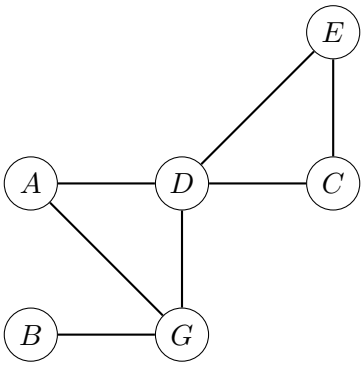
   Here, "alphabetical order" refers to standard English ordering of letters: ABCDEFGHIJKLMNOPQRSTUVWXYZ

   (a) Here is a binary heap, draw as a tree. What is the character array/string representation of the heap? Write only the seven characters in the form.



   (b) Suppose we perform a `remove-min` operation on this heap. What is the resulting array representation? You may wish to do your work in terms of the heap drawn as a tree, above. Write only the six characters in the form.

4. (1 point) Suppose we run a valid breadth-first order of the following graph, starting at vertex A. This means that vertex A is in layer zero on its own.

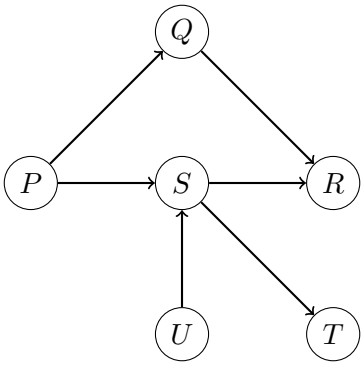


   (a) In which layer is vertex E?

   (b) In which layer is vertex C?

   (c) In which layer is vertex G?

5. (1 point) Give a valid topological order of the following graph. Write only the six letters in the short-response answer form.



6. (0.5 points) This question deals with `InsertionSort`, the relevant code of which is reproduced for your convenience:

```
for j ← 2 to n do
    key ← A[j]
    i ← j − 1
    while i > 0 and A[i] >key do
        A[i + 1] ← A[i]
        i = i − 1
    A[i + 1] ← key
```

The following array is in the middle of being sorted by InsertionSort. Right now, the next line of code to execute is for j to be incremented.

16	23	52	26	92	53	32	10	14	58	79	19	82	98	57

What is the maximum value that j can have right now (before the increment)? Recall that when we discuss pseudo-code, we treat arrays as indexed $1 \ldots n$, so if your belief is that the outer loop has executed exactly once, your answer should be two.

7. (0.25 points) This question deals with `SelectionSort`, the relevant code of which is reproduced for your convenience:

```
for i ← 1 to n − 1 do
    min ← i
    for j ← i + 1 to n do
        if A[j] < A[min] then
            min ← j
    Swap A[i] and A[min]
```

The following array is in the middle of being sorted by SelectionSort. Right now, the loop condition for the outer loop (for loop indexed by i) is about to be checked.

10	20	30	70	45	32	78	90

What is the maximum number of times the loop could have executed so far?

8. (0.25 points) This question deals with `BubbleSort`, the relevant code of which is reproduced for your convenience:

   for $i \leftarrow 1$ to $n-1$ **do**
     for $j \leftarrow 1$ to $n-i$ **do**
       if $A[j+1] < A[j]$ **then**
         Swap $A[j]$ and $A[j+1]$

   The following array is in the middle of being sorted by BubbleSort. Right now, the next line of code to execute is for i to be incremented.

16	23	52	26	53	32	10	14	58	79	19	82	57	92	98

   What is the maximum value that i can have right now (before the increment)? Recall that when we discuss pseudo-code, we treat arrays as indexed $1 \ldots n$.

9. (0.5 points) This question deals with `HeapSort`. Recall that this algorithm converts a vector into a max heap and then performs $n-1$ instances of the `extractMax` procedure.

   The following array is in the middle of being sorted by HeapSort. We are already past the portion of the code that converts the vector to a max heap. We may have performed some number of the extractMax procedure.

82	79	57	58	53	52	10	14	26	23	32	16	92	98

   How many times has extractMax been performed?