1. For 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.

   $a(n) = 1729n$ ☐ $\mathcal{O}(n)$ ☐ $\Omega(n)$
   $b(n) = n^{\log n}$ ☐ $\mathcal{O}(n)$ ☐ $\Omega(n)$
   $c(n) = \sqrt{n}$ ☐ $\mathcal{O}(n)$ ☐ $\Omega(n)$

   $\sqrt{n}$ is $\mathcal{O}(n)$ but not $\Omega(n)$. $n^{\log n}$ is Ω but not $\mathcal{O}(n)$ $1729n$ is both.

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

   For the ordering requirement, write the function identifier **letters** in order.

   The value e refers to the base of the natural logarithm; $e \approx 2.718281828$

   (a) $a(n) = n + n^2 + \sqrt{n}$
   (b) $b(n) = \log^3 n + \log \log n$
   (c) $c(n) = 2^{n \log n}$
   (d) $d(n) = n^e$
   (e) $e(n) = (7^{e-1/e} - 9) \cdot \pi^2$
   (f) $f(n) = 2^n$
   (g) $g(n) = \sqrt{\log n}$
   (h) $h(n) = 4^{\log n}$

   The correct ordering is

   $egba = hdfc$

   Also ah can be in either order since they are Θ of one another

3. 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 being sorted by SelectionSort. Right now, the loop condition for the outer loop (for loop indexed by i) is about to be checked.

30	39	87	96	56	84	46	58	43	55

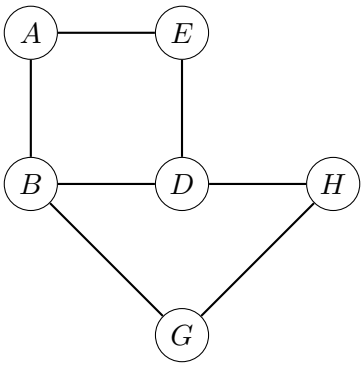
(a) What is the minimum number of times the loop could have executed so far? Briefly justify (1-2 sentences) your answer.

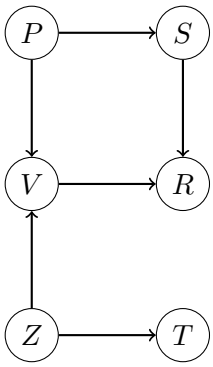
Minimum zero. The array could have started this way.

(b) What is the maximum number of times the loop could have executed so far? Briefly justify (1-2 sentences) your answer.

Two. The first two elements of the vector have the correct elements, but the third does not. The first two could be there by coincidence or have started there, or they could be there because of iteration(s) of the algorithm. However, if there were three iterations, then the third spot would have 43 instead of 87.

4. Give a valid breadth-first order of the following undirected graph, starting at vertex A.





5. Give a valid topological order of the above **directed** graph.

For questions 4 and 5, be sure to use the correct graph. If you aren't sure what an undirected or directed graph is, review the video before answering the questions.

There are many correct answers to these two questions.

One such example would be:

Breadth-first: AEBDGH

Topological: PZVSRT