When measuring the time efficiency of an algorithm, why not just use a clock to see how long it takes to run?

A. Time efficiency is measured by how much space an algorithm takes to execute.

B. The processors on different computers would result in different time measurements for a given algorithm.

C. A clock is not accurate enough to measure the efficiency of an algorithm.

D. The memory on a computer is a better way to measure time efficiency.

Answer Point Value: 5.0 points
Answer Key: B
2. What is meant by the “worst case input” for an algorithm when analyzing it for time efficiency?

- A. The amount of time that an algorithm takes to execute.
- B. The input that would make the algorithm do the most work.
- C. The worst implementation of an algorithm to solve a problem.
- D. The input that is used when an algorithm needs to be tested.

Answer Point Value: 5.0 points
Answer Key: B

3. Suppose you have the following Python function that returns all of the readings from a sensor that are over a certain threshold:

```python
def getHighReadings(readings, threshold):
    highReadings = []
    for i in range(len(readings)):
        if readings[i] > threshold:
            highReadings = highReadings + [readings[i]]
    return highReadings
```

What is the worst case input for this algorithm?

- A. 100
- B. Linear.
- C. \(O(n^2)\)
- D. All readings are above the threshold.

Answer Point Value: 5.0 points
Answer Key: C
Which of the following is true about order of magnitude or "order notation" with respect to analysis of algorithms? (Choose all that apply)

A. An algorithm with $O(n)$ amount of work is more efficient than an algorithm with $O(n^2)$ to solve the same problem.

B. It is used to compare the amount of work that different algorithms must perform to do the same job.

C. It refers to how many items a sorting algorithm can sort.

D. It measures the total amount of work done by an algorithm.

Answer Point Value: 5.0 points
Answer Key: A,B