

What is the growth order of the following code? What is the time complexity?

```
for(int j=0; j<V; j++)  
    sum++;
```

```
for(int k=0; k<V; k++)  
    sum2++;
```

- POWERUNIT
- Growth order = $4V$, $-(4V - 4)$
 - Growth order = $V - 2$, $-4V$
 - Growth order = V , $-4V$
 - Growth order = V^2 , $-4V^2$
 - Growth order = $4V$, $-4V$

[Clear my choice](#)

Assume during the experimental analysis of a code. The result of the log-log plot with slope equal 4. If the execution time = 2 seconds for input size 100,000, what is the execution time for input size 200,000?

- 6 seconds
- 8 seconds
- 10 seconds
- 16 seconds
- 2 seconds

Clear my choice

POWERUNIT

```
Public class GeneralClass {  
    private int X;  
    private double Y;  
    private double W;  
    private Char Z;  
}
```

What is the size of the object of type GeneralClass which is shown above?

- 21 Bytes
- 37 Bytes
- 40 Bytes
- 22 Bytes
- 38 Bytes

[Clear my choice](#)

inserting an element in a resizing array stack is called _____

enqueue

Create



Push

Pop

Evaluation

[Clear my choice](#)

Given that we built a stack using resizing array that is doubled when full, and using an initial size = 5 elements, how many array accesses will be needed to push 50 elements?

None of the answers is correct

150

200

120

125

[Clear my choice](#)



Which of the following covers the 'Average' case scenario?

- Big (θ)
- Big (Ω)
- All of the mentioned
- Big (O)
- Telda Approximation

[Clear my choice](#)

What is the time and memory complexities of following code:

```
int a = 0, b = 0;  
for (i = 0; i < N; i++)  
    a = a + sqrt(a);  
for (j = 0; j < M; j++)  
    b = b + sqrt(a);
```

$O(N + M)$ time, $O(N + M)$ memory

$O(N + M)$ time, $O(1)$ memory

$O(N * M)$ time, $O(1)$ memory

$O(N * M)$ time, $O(N + M)$ memory

None of the answers is correct

[Clear my choice](#)

In quicksort, after finishing the first partitioning, the array is looking like this:

2 5 1 9 11 14 13 12

Which statement is correct?

- The pivot could be either the 9 or the 11.
- Neither the 9 nor the 11 is the pivot.
- The pivot could be the 9, but it is not the 11
- We can't tell any conclusion about the possible pivot
- The pivot is not the 9, but it could be the 11

Clear my choice

Given that the median of a set of elements is the element in the middle where half of elements are bigger than the median and half are smaller.

Suppose to find the median of an unsorted array, we need $O(n)$. Now consider a QuickSort implementation where we first find median using the above algorithm, then use median as pivot. **What will be the worst case time complexity of this modified QuickSort.**

- $O(n^2 \log n)$
- $O(n \log n \log n)$
- $O(n^3)$
- $O(n \log n)$
- $O(n^2)$

POWERUNIT

time
Consider the following pseudocode that uses a stack. What is output for wordX = "ABCDEFGHijkl"?

```
declare a stack of characters
while (there are more characters in the wordX)
{
  read a character
  push the character on the stack
  if (the stack has 4 characters) {
    while (the stack is not empty) {
      pop a character off the stack
      write the character to the screen
    }
  }
}
```

Answer: DCBAHGfELKJ|

What is the time complexity of following code:

```
int a = 0, i = N;  
while (i > 0) {  
    a += i;  
    i /= 2;  
}
```

- $O(N/2)$
- $O(\text{sqrt}(N))$
- $O(N)$
- $O(\log_2 N)$
- $O(N^2)$

[Clear my choice](#)

Which of the following is not an in-place sorting algorithm?

- None of the answers
- Quick sort and Insertion sort
- Merge sort
- Selection sort and merge sort
- Insertion sort and merge sort

[Clear my choice](#)

The worst case occurs in quick sort when -----

- The pivot is a random element
- Pivot is the median of the array
- None of the mentioned
- Pivot is the largest element
- Pivot is the middle element

[Clear my choice](#)

Consider the following operation performed on a stack of size 4.

Push(1);

Pop();

Push(2);

Push(3);

Pop();

Push(4);

Pop();

Push(5);

Push(6);

Push(7);

POWERUNIT

After the completion of all operations, the number of elements present in stack is?

2

4

3

1

5

In resizing array stack, if somebody is inserting an element into stack already having five elements and stack size of 5, then stack becomes

- Garbage Collection
- Underflow
- Overflow
- pushing a null element
- Empty collection

[Clear my choice](#)



Time left 0:48:27

Given that the median of a set of elements is the element in the middle where half of the elements are bigger than the median and half are smaller.

Suppose to find the median of an unsorted array, we need $O(n^2)$. Now consider a QuickSort implementation where we first find median using the above algorithm, then use median as pivot. What will be the worst case time complexity of this modified QuickSort.

- $O(n \log n \log n)$
- $O(n^2 \log n)$
- $O(n^2)$
- $O(n \log n)$
- $O(n^3)$

Reading an element from a Linked list stack is called _____

- Evaluation
- Pop
- Push
- enqueue
- Create

Clear my choice

Consider the following pseudocode that uses a stack. What is output for **wordX** = "ABCDEFGHijkl"?

```
declare a stack of characters
while (there are more characters in the wordX)
{
  read a character
  push the character on the stack
  if(the stack has 3 characters){
    while(the stack is not empty){
      pop a character off the stack
      write the character to the screen
    }
  }
}
```



Answer:

To sort an array of m items using insertion sort, the worst case complexity is _____

$O(m \log m)$

$O(m^2)$

$O(\log m)$

$O(m)$

$O(m^2 \log m)$

[Clear my choice](#)

POWERUNIT

What is the time and memory complexities of

```
int a = 0, b = 0;
for (i = 0; i < V; i++)
    a = a + rand();
for (j = 0; j < W; j++)
    b = b + rand();
```

$O(V + W)$ time, $O(1)$ memory

None of the answers is correct

$O(V + W)$ time, $O(V + W)$ memory

$O(V * W)$ time, $O(1)$ memory

$O(V * W)$ time, $O(V + W)$ memory

Clear my choice

time left 0:25

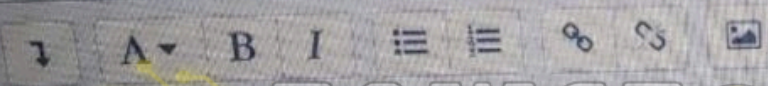
Given that there is a stack that is built using resizing array. The size is multiplied by 3 when it is full and size is divided by 3 when number of elements = $1/6$ length. Using shallow analysis where we assume each stack element is 8 Bytes only, what is the total approximation of the size reserved for this stack assume it has N elements?

- 8N to -48N
- 32N to -48N
- 8N to -32N
- 8N to -24N
- None of the mentioned

Clear my choice

POWERUNIT

What is the advantage of a stack using resizing array compared to a stack using linked list?



POWERUNIT

If the number of records to be sorted is small, then sorting can be eff

- Selection
- Merge
- Either Insertion or Merge
- Either Selection or Merge
- Either Merge or Quick

Clear my choice

POWERUNIT

To sort an array of n items using insertion sort, the worst case complexity is -----

$O(\log n)$

$O(n)$

$O(n^2)$

$O(n \log n)$

$O(n^2 \log n)$

[Clear my choice](#)

POWERUNIT

If the number of records to be sorted is small, then sorting can be efficient.

- Selection
- Either Merge or Quick
- Merge
- Either Selection or Merge
- Either Insertion or Merge

[Clear my choice](#)

POWERUNIT

Assume that the exchange operation is very costly. Which of the following sorting algorithms will minimize the number of exchange operations in general?

- Insertion Sort
- Quick Sort and selection sort
- All of them take the same number of exchanges
- Selection Sort

Quick sort

[Clear my choice](#)