find BigO for all below

1. int thrower(){

int arr[5]={1000, 2, 3, 17, 50}; O(5)

return arr[0]=25; O(1)

}

**Total=O(5)+O(1)=O(5)**

2. int container(int n) {

for(int i=0;i < n;i++){ O(n)

if(i%2){ O(n)

return true; O(1)

}

}

**Total=O(n).O(n)+O(1)=O(n2)**

3. int naiveFib(int n){

for(int i=0;i < n;i++){ O(n)

for(j=0;j< n;i++){ O(n)

if(i%2==0){ O(n)

return true; O(1)

}

}

}

**Total=O(n).O(n).O(n).O(1)=O(n3)**

4. unsigned int naiveFib(int n){

if(n==0){ O(1)

return 0; O(1) }

else if(n==1){ O(1)

return 1; O(1)

}

else{

return ( naiveFib(n-1) + naiveFib(n-2) ); O(n)

}

}

**Total= O(1)+O(1)+O(1)+O(1)+O(n)=O(n)**

**Answer some of the questions below**

**1. How Stack is different from Array?**

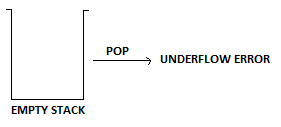
A stack is a last in, first out (LIFO) abstract data type and linear data structure while an array is a data structure consisting of a collection of elements (values or variables), each identified by at least one array index or key.

In array you can access any elements of array at any time but in stack, there's no random-access operation; there are only Push, Peek and Pop, all of which deal exclusively with the element on the top of the stack.

As we know array can be one dimensional and two dimensional but stack is only one dimensional.

**2. What do you mean by stack underflow and stack overflow? Explain with some diagrammatic representation.**

Stack underflow is the error caused when we try to remove an element from stack when there is nothing to remove.



Stack overflow is the error caused when we try to add an element to stack when the stack is already full.

