

1. Write the time complexity of each of the code segments shown below.

•

```
int k=1;
while(k<=n){
    cout<<k<<endl;
    k=k*2;
}
```

Solution: $\log_2 n$

•

```
for (int i = 0; i < n; i++)
    for (int j = i; j > 0; j--)
        cout << i << j;
```

Solution: n^2

•

```
for (int i = 0; i < n; i++)
    for (int j = i; j > 0; j--)
        for(int k=j; k > 0; k--)
            cout << i << j << k;
```

Solution: n^3

•

```
for(int i=n/2;i<=n;i++){
    for(int j=1;j<=n;j=j*2){
        cout<<i<<j<<endl;
    }
}
```

Solution: $n \log_2 n$

•

```
for(int i=n/2;i<=n;i++){
    for(int j=1;j<=n;j=j+i){
        cout<<i<<j<<endl;
    }
}
```

Solution: $n \log n$

•

```
for(int i=1;i<=n;i++)
{
```

```

        if(builtin_popcount(i) == 1)
        {
            for(int j=1;j<=n;j++)
                cout<<i<<j<<endl;
        }
    }

```

Note: builtin_popcount(i) returns the number of set bits in 'i'.
For example, builtin_popcount(5) = 2. Because, $5 = (101)_2$. So there are 2 set bits in 5.

Solution: $n \log_2 n$

2. WAP that takes an array of size n and q queries as input. For each query you will be given a number. For each query you have to print 'YES' if the number is present in the array, otherwise print 'No'. Solve the problem in $O((n+q) \log n)$

Sample input	Sample output
5	YES
6 3 2 1 8	NO
4	YES
1	NO
5	
2	
9	

Explanation: You are given an array of size 5 and 4 queries. In the first query you are given 1. 1 is present in the array so we print 'YES'. In the second query 5 is not present in the array so we print 'NO'. Third and Fourth query are similar.

Hint: First sort the array with the STL sort function which works in $O(n \log n)$, then for each query binary search the value in the sorted array.

3. Modify the bubble sort and insertion sort code to sort in non-increasing order.

Sample input	Sample output
5	5 4 4 2 1
4 5 2 4 1	
3	2 1 1
1 2 1	