CS211 Quiz_1 (471)

Q1/ Determine the complexity (Big O) of the following code:

```
void merge_two_arrays(int arr1[], int n1, int arr2[], int n2) {
 int i =0, j=0;
 while (i<n1 && j<n2) {
  if (arr1[i] < arr2[j]) {
   printf("%d ", arr1[i]);
   i++;
  } else {
   printf("%d", arr2[j]);
  }
 }
 j++;
 while (i<n1) {
  printf("%d", arr1[i]);
  i++;
 }
 while ( j<n2) {
  printf("%d", arr2[j]);
  j++;
 }
}
```

a) O(n1∗n2) b) O(n2) c) O(n1 + n2) d) O(n1)

Q2/ Determine the complexity (Big O) of the following code:

```
int find_target(int arr[ , int n, int target)
{for ( int i=0; i<n; i++) {
  if (arr[i]== target) {
   return 1; // Target found
  }
 }
 return 0; // Target not found
}
```

a) O(n log(n)) b) O(1) c) O(log(n)) d) O(n)

Q3/ Determine the complexity (Big O) of the following code:

```
int find_max(int arr  , int n) {
 int max = arr[0];
 for (int i = 1; i < n; i++) {
  if ( (arr[i]>max) {
   max = arr[i];
  }
 }
 return max;
}
```

a) O(n) b) O(n log(n)) c) O(1) d) $O(n^2)$

Q4/ Determine the complexity (Big O) of the following code:

```
int access_element(int arr[], int index) {
}
 return arr[index];
```

a) O(n) b) O(log(n)) c) O(1) d) $O(n^2)$

Q5/ Determine the complexity (Big O) of the following code:

```
int bs(int arr  , int n, int target) {
 int low =0; high = n - 1;
 while (low <= high) {
  int mid = (low + high) / 2;
  if (arr[mid] == target) {
   return mid;
  } else if (arr[mid] < target) {
   low = mid + 1;
  } else {
   high = mid - 1;
  }
 }
 return -1;
}
```

a) O(n) b) O(log(n)) c) O(1) d) O(n log(n))

Q6/ Select the lowest complexity:

a) Cubic b) Quadratic c) Constant d) Linear

Q7/ Determine the complexity (Big O) of the following code:

```
void print_triplets(int arr[], int n) {
 for ( int i=0; i < n; i++) {
  for ( int j=0; j<n; j++) {
   for (int k=0; k < n; k++) {
   }
  }
 }
 printf("(%d, %d %d)\n", arr[i], arr[j], arr[k]);
}
```

a) O(n log(n)) b) $O(n^3)$ c) O(n) d) $O(n^2)$

Q8/ Determine the complexity (Big O) of the following code:

```
void b_sort(int arr  , int n) {
 for (int i=0; i<n-1; i++) {
  for (int j=0; j<n-i-1; j++){
   if (arr[j]>arr[j+1]) {
   }
   int temp=arr[j];
   arr[j]=arr[j+1];
   arr[j+1]=temp;
  }
 }
}
```

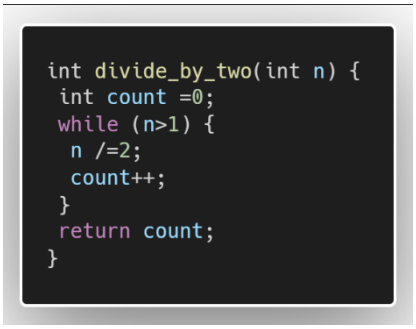
a) O(n) b) $O(n^2)$ c) O(log(n)) d) O(1)

Q9/ Determine the complexity (Big O) of the following code:

```
void f(int n) {
 int i=0;
 while (i<n) {
  i+=1;
  printf("%d\n", i);
 }
}
```

a) $O(n^2)$ b) O(log(n)) c) O(1) d) O(n)

Q10/ Determine the complexity (Big O) of the following code:



```
int divide_by_two(int n) {
 int count =0;
 while (n>1) {
  n /=2;
  count++;
 }
 return count;
}
```

a) O(n) b) $O(n^2)$ c) O(log(n)) d) O(1)

Answers on the next page

Answers:

Q1	C	**Q6**	C
Q2	D	**Q7**	B
Q3	A	**Q8**	B
Q4	C	**Q9**	D
Q5	B	**Q10**	C

"Made by Omar Edilbi"