

1. Give the big O for the following algorithms. Circle the appropriate time complexity with respect to N.

1A.

<pre>int A (int N) { int i, j, k= 35; for (j=0;j<k;j++){ for(i=0;i<N*N;i++) { j= j*k/3; } } }</pre>	O(N)	O(1)
	O(NlogN)	O(logN)
	O(N ²)	O(2 ^N)
	Infinite	None of
		the above

1B.

<pre>int B(int N) { B(N-1); if(N<=0) return 1; }</pre>	O(N)	O(1)
	O(NlogN)	O(logN)
	O(N ²)	O(2 ^N)
	Infinite	None of
		the above

1C.

<pre>void C(int N) { if(N<=0) return 1; else { C(N-1); C(N-1); } }</pre>	O(N)	O(1)
	O(NlogN)	O(logN)
	O(N ²)	O(2 ^N)
	Infinite	None of
		the above

1D.

<pre>int D (int N) { int i, j, k=0, x; for (i=0;i<N;i++) { k=k+1; } for (j=0;j<k;j++){ for(i=0;i<N;i++) { for(x=0;x<k;x++) { x = k; } } } return x; }</pre>	O(N)	O(1)
	O(NlogN)	O(logN)
	O(N ²)	O(2 ^N)
	Infinite	None of
		the above

2. Consider the following implementation of Quicksort.

```
void Partition (int A[], int *i, int *j) {
    int pivot, temp, pivpos;
    pivpos=(*i+*j)/2 ;
    pivot = A[ pivpos ]; //middle key
    do {
        while ( A[*i] < pivot ) (*i)++;
        while ( A[*j] > pivot ) (*j)--;
        if (*i <= *j ) {
            temp = A[*i]; A[*i]=A[*j]; A[*j]=temp; //swap i, j
            (*i)++; (*j)-- ;
        }
    } while (*i <= *j);
}

void QuickSort (int A[], int m, int n) {
    int i,j;
    if (m<n) {
        i=m; j=n;
        Partition (A, &i, &j);
        QuickSort (A,m,j);
        QuickSort (A,i,n);
    }
}
```

Suppose QuickSort(A, 0, 7) was called where A[] = {4, 13, 1, 7, 9, 3, 16, 23}

- 2A. What is the array after the FIRST call to Partition returns?
- 2B. What is the array after the SECOND call to Partition returns?
3. Given the following array A[] = { 2,7,26,25,19,17,1,90,3,36}.
- 3A. Draw a MAX Binary Heap Tree.
- 3B. What does the Heap Tree look like after the 2 rounds of extracting the largest element?
4. Hashing
- 4A. Using the division hash function: $\text{hash}(k) = k \% 4$. Insert the following key: 1, 4, 5, 2, 3, 6 into a chained hash table of size 6. Show all your work for each insertion
- 4B. Is the hash function in 7A a perfect hash? Why or why not?
- 4C. Rewrite the hash function to improve it. Insert the same keys using your new hash function into a chained hash table of size 6. Show all your work for each insertion