**Complexity Analysis**

Q1. Arrange the following in **increasing** order of complexity

2n, n\*log(n), n2, n, n2logn, 7, log(log n)

Q2. What is the time complexity (worst case) of the following code:

1. void foo(int n) {

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

for (int j = 1; j < i; j++)

printf(“%d\n”,i);

}

1. void foo(int n) {

for (int i = 1; i < =n/2; i++)

for (int j = 1; j <= n; j=j+2)

printf(“%d\n”,i);

}

1. void foo(int n) {

for (int i = n/2; i < n; i++)

for (int j = 1; j <n; j= j\*2)

printf(“%d\n”,i);

}

1. void foo(int n) {

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

for (int j = n; j > 1; j=j/2)

for (int k = 1; k < j; j++)

printf(“%d\n”,i);

}

1. void foo(int n) {

for (int i = n/2; i < n; i=i\*2)

printf(“%d\n”,i);

for (int i = n/2; i < n; i++)

for (int j = 1; j < i; j++)

printf(“%d\n”,i);

}

1. int fun1 (int n){

int i, j, k, p, q = 0;

for (i = 1; i < n; ++i)

{

p = 0;

for (j = n; j > 1; j = j/2)

++p;

for (k = 1; k < p; k = k\*2)

++q;

}

return q;

}

1. void foo(int n, int m) {

for (int i = n/2; i <= n; i++)

printf(“%d\n”,i);

for (int i = 1; i <= m; i++)

printf(“%d\n”,i);

}

1. void foo(int n) {

for (int i = 2; i <= n; i= pow(i,c) )

printf(“%d\n”,i);

}

1. int search(int a[],int n, int x){

int i, flag =0;

for ( i=0;i<n && !flag ;i++)

if (a[i] == x) flag =1;

if (i==n) return -1;

else return i;

}