Foundations of Algorithms and Machine Learning

(CS60020)

Spring Semester 2017

Assignment - 2

1. Calculate the best case and worst case time complexity of Bubble sort algorithm.

```
BUBBLESORT(A)
1. for i ← 1 to length[A]
2.       do for j ← length[A] downto i + 1
3.             do if A[j] < A[j - 1]
4.                   then exchange A[j] ↔ A[j - 1]
```

2. What is the worst case time complexity of the following:

```
int fun(int n)
{
  int count = 0;
        for (int i = n; i > 0; i /= 2)
              for (int j = 0; j < i; j++)
                    count += 1;
  return count;
}
```

3. Prove or Disprove the following:

(a) Is $2^{n+1} = O(2^n)$?

(b) Is $2^{2^n} = O(2^n)$?

(c) Is $4^{\lg n} = \Omega(n^3)$?

4. Calculate the Worst case time complexity of the following code:

```
for (int i = 1; i <=n; i += 1)
{
        for (int j = 1; j <=n; j += 1)
        {
              print("hi");
        }
}

for (i = n; i > 0; i /= 2)
{
     print("hello");
}
```

5. Find the average case time complexity of binary search algorithm:

```
function binsrch2(low, high: index)

if low > high then return(0)
else
        mid ← ⌊(low + high)/2⌋
        if x = A[mid] then
            return(mid)
        else if x < A[mid]
            return(binsrch2(low, mid - 1))
        else
            return(binsrch2(mid + 1, high)
endif
```