UWI

Design and Analysis of Algorithms

Tutorial 6

Setting Up & Solving Recurrence Relations

1. Setup and solve the recurrence relation for the following algorithm.

```
Algorithm F(n)
if n = 0 return 1
else return F(n - 1) × n
```

2. Setup and solve the recurrence relation for the following algorithm.

```
Algorithm S(n)
//Input: A positive integer n
//Output: The sum of the first n cubes
if n = 1 return 1
else return S(n - 1) + n * n * n
```

3. Setup the recurrence relation for the following algorithm.

```
Algorithm BinarySearch(A[0...n - 1], K)
//Implements non-recursive binary search
//Input: An array A[0...n - 1] sorted in ascending order and a search key K
//Output: Index of the array's element that is equal to K or -1
l = 0;
r = n - 1
while l <= r do
    m = floor((l + r)/2)
    if K = A[m] return m
    else if K < A[m] r = m - 1
    else l = m + 1
return -1
```

Transform and Conquer Approach

It is important that we learn the theory behind the transform and conquer approach and its algorithms.

1. Explain the theory behind the transform and conquer approach
   (a) What is meant by the transform and conquer approach?
2. Explain the theory behind heaps and heap sort.
   (a) What is a heap?
   (b) What is the heap property?

(c) What are the two stages of Heap sort?

(d) What is the overall time complexity of heap sort?

(e) Calculate the time complexity of the worst case scenario of the heapBottomUp operation, given that: