Lecture 26: Analyzing Divide and Conquer Algorithms

Date: November 4, 2019.

```
MergeSort(A[l .. u])
    if n > 1
        m = ⌊ (u-l)/2 ⌋
        MergeSort(A[l .. m])
        MergeSort(A[m+1 .. u])
        Merge(A[l .. u], m)
```

```
Merge(A[l .. u], m)
    i = l; j = m+1
    for k = l to u
        if j > u
            B[k] = A[i]; i = i+1
        else if i > m
            B[k] = A[j]; j = j+1
        else if A[i] < A[j]
            B[k] = A[i]; i = i+1
        else
        B[k] = A[j]; j = j+1
    for k = l to u
        A[k] = B[k]
```

Problem 1. What is the running time of the above algorithm?

```
BinarySearch(A[l ..  u], x)
    if (u - l < 0) return NO
    mid = ⌊l+u/2⌋
    m = A[mid]
    if (x = m) return YES
    else if (x < m)
        return BinarySearch(A[l ..  mid-1],x)
    else
        return BinarySearch(A[mid+1 ..  u], x)
```

Problem 2. What is the running time of the above algorithm?

Problem 3. Suppose the running of an algorithm is given by $T(1) = 1$ and

$$T(n) = T(3n/4) + T(n/4) + n$$