Day 10

ANALYSIS PATTERNS

QUICKSORT

MERGESORT

Linear Search

```
public static int linearSearch(int[]a, int e){
    for (int i = 0; i < a.length; i++){
        count++;
        if (a[i] == e) return i;
    }
    return -1;
}
```

Best case: O(1)
Worst case: O(N)
Average case: $O(\log_2 N)$

Linear Search

```
public static int linearSearch(int[]a, int e){
    for (int i = 0; i < a.length; i++){
        count++;
        if (a[i] == e) return i;
    }
    return -1;
}
```

Average case analysis.

Probability: 1/N 1/N 1/N 1/N 1/N 1/N 1/N 1/N 1/N 1/N

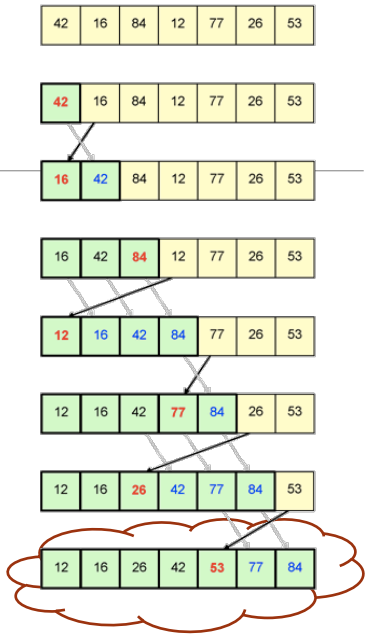
Num comp: 1 2 3 4 5 6 7 8 9 10

Grand total:

$$\sum_{i=1}^{n} 1/n * i = 1/n * \sum_{i=1}^{n} i = 1/n * (n * (n+1))/2 = (n+1)/2$$

Insertion Sort

ALGORITHM *InsertionSort*$(A[0..n-1])$
//Sorts a given array by insertion sort
//Input: An array $A[0..n-1]$ of n orderable elements
//Output: Array $A[0..n-1]$ sorted in nondecreasing order
for $i \leftarrow 1$ **to** $n-1$ **do**
  $v \leftarrow A[i]$
  $j \leftarrow i-1$
  while $j \geq 0$ **and** $A[j] > v$ **do**
    $A[j+1] \leftarrow A[j]$
    $j \leftarrow j-1$
  $A[j+1] \leftarrow v$

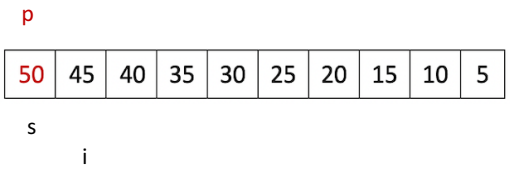


Tracing Lomuto Partitioning

```
LomutoPartition(A[l..r])
    p ← A[l]
    s ← l
    for i=(l+1)..r
        if A[i] < p
                increment s
                swap(A[s],A[i])

    swap(A[l],A[s])

    return s
```



5