Lab 2 (17 Jan 2019)

Problem 1 [Tower of Hanoi]: Write a program to solve the tower of Hanoi problem. Your program should print the sequence of moves required to move the discs from the source peg to the target peg. Try running your program for increasingly large values of n and plot the change in your program's run time.
Sample Input/Output:

```
Input:
Enter the number of disks: 3

Output:
Move disc 1 from S to T
Move disc 2 from S to I
Move disc 1 from T to I
Move disc 3 from S to T
Move disc 1 from I to S
Move disc 2 from I to T
Move disc 1 from S to T
```

Problem 2: Write a program to print the *peak element* in a sequence of integers that first strictly increase and then strictly decrease. For e.g. for the sequence 2,4,6,8,7,5,3 your program should print 7 and for 10, 12, 8, 4 , -3, -15 it should print 12. Your program should run **asymptotically faster** than linear time. (*Hint: Think of a O(logn) time algorithm. How to achieve such a runtime?)*

Problem 3: An *inversion* in a sequence A of numbers is a pair of indices (i,j) such that i<j and A[i]>A[j]. Write a program to count the total number of inversions in an input sequence. For e.g. the number of inversions in 1,3,9,8,5 is 3 while that in 4,10,8,2,1 is 8. [*Write a simple naive algorithm to do this. What is its runtime? Can you do better?*]