DSC 40B - Discussion 03

Problem 1.

a) State (but do not solve) the recurrence relation describing this function's run time.

```
import random
def foo(n):
    if n <= 2:
        return
    for i in range(n):
        for j in range(i,n):
            print(i)
    return foo(n//2) + foo(n//2)
```

b) Suppose a binary search is performed on the following array using the implementation of *binary_search* from lecture. What is the worst case number of comparison's that would be made to search for an element in the array.

```
[1, 4, 7, 8, 8, 10, 15, 51, 60, 65, 71, 72, 101]
```

Problem 2.

We're given two lists, `A` and `B` and a target `t`, and our goal is to find an element a of `A` and an element b of `B` such that $a + b = t$.

Problem 3.

Solve the following recurrence relations.

a) T(n) = T(n-1) + n
T(0)=0

b) T(n)=4T(n/4) + n
T(1)=1

Problem 4.

Determine the recurrence relation describing the time complexity of each of the recursive algorithms below.

a)
```
def fact(n):
    if(n <= 1)
        return 1
    else
        return n*fact(n-1)
```

b)
```
def max_arr(arr):
    if(len(arr) == 1):
        return arr[0]
    mid = len(arr)//2
    left_max = max_arr(arr[:mid])
    right_max= max_arr(arr[mid:])
    if(left_max>right_max):
```

```
        return left_max
    else:
        return right_max
```