Problem Set 1

Induction

1.)

Find the closed form of(via math or online resource):

$$\sum_{i=1}^{n} 4^i$$

Prove your closed-form formula via induction.

2.) Analysis

```
i <-- n;
while(i > 1) {
  j = i;      //%% CAUTION: this DOES NOT START AT 0
  while (j < n) {
    k <-- 0;
    while (k < n) {
      k = k + 2;
    }
    j <-- j * 2;
  }
  i <-- i / 2;
}
```

What is the asymptotic upper bound of the code above?

3.)Analysis

```
float useless(A){
  n = A.length;
  if (n==1){
    return A[0];
  }
  let A1,A2 be arrays of size n/2
  for (i=0; i <= (n/2)-1; i++){
    A1[i] = A[i];
    A2[i] = A[n/2 + i];
  }
  for (i=0; i<=(n/2)-1; i++){
     for (j=i+1; j<=(n/2)-1; j++){
        if (A1[i] == A2[j])
           A2[j] = 0;
     }
  }

  b1 = useless(A1);
  b2 = useless(A2);
  return max(b1,b2);
}
```

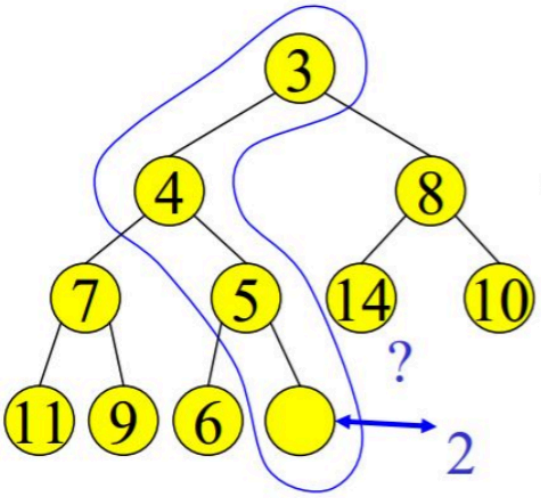
What recurrence equation describes the code above?

4.) MinHeap Review

If you haven't covered heaps please review here:

https://youtu.be/WCm3TqScBM8?si=9XIQqe9qB88OtRXS

a.) See the **min**Heap below. Please note that in this structure the parents are equal or smaller than their children. Show the resulting tree after you push(2), push(31), pop(), pop(), and update the key of 7 to -2.



b.) Given the following functions *push(a[1])*, *push(a[2])*, *push(a[3])* … *push (a[n])* The asymptotic upper bound on loading the heap assuming the heap is initially empty is O(n). This video gives a formal explanation. Can you try to explain why the runtime is O(n) in your own words?https://youtu.be/MiyLo8adrWw?si=BI7_LCyoQX1oqg6S

5.)Analysis

Given an analysis of the running time in Big-O for problems a-f

(a)
```
sum = 0;
  for( i = 0; i < n; ++i )
                ++sum;
```

(b)
```
sum = 0;
for( i = 0; i < n; ++i )
        for( j = 0; j < n; ++j )
                ++sum;
```

(c)
```
sum = 0;
 for( i = 0; i < n; ++i )
  for( j = 0; j < n * n; ++j )
           ++sum;
```

(d)
```
sum = 0;
 for( i = 0; i < n; ++i )
    for( j = 0; j < i; ++j )
             ++sum;
```

(e)
```
sum = 0;
  for( i = 0; i < n; ++i )
          for( j = 0; j < i * i; ++j )
            for( k = 0; k < j; ++k )
                        ++sum;
```

(f)
```
sum = 0;
     for( i = 1; i < n; ++i )
           for( j = 1; j < i * i; ++j )
                 if( j \% i == 0 )
                          for( k = 0; k < j; ++k )
                                ++sum;
```

6.)

Is $\log_4 n = O(\log_{16} n)$? What about $\log_{16} n = O(\log_4 n)$? Why or why not?

7.a)

Rank the following time bounds. That is write them as $f_1, f_2, ..., f_6$ and show that $f_i = O(f_{i+1})$ for all $1 \leq i \leq 5$ (You may use limit lemma theorem)

- $3n^4 + 6n$
- $n \log(n^{1000})$
- $7n^3 \log(n) + 1000$
- 3^n
- 6^n
- $1024n^2 + 4n + 460$

1

7b.) Prove that $k(n) = n^2 + 3n$ is $\Omega(n^2)$.

8.)

a.i)Given S={z, b, g}
List all subsets of S.

a.ii) Given S={z, b, g, r}
List all subset of S

b.) List all substrings of cdef

c.) Given S=100, How many subsets can be created?

d.) Create a pseudocode for the following function decode(x)

Input: Int x, set S of size n, where $x < (2^n - 1)$
Output: the dubset represented by the binary digit

Example1 Input: decode(7) S=A,B,C
Because binary(7)=111 that means we include all members of set S
Output:{A,B,C}

Example2 Input: decode(5) S=A,B,C
Because binary(5)=101 that means we include first and last member
Output:{A,C}

e.) Provide pseudocode the creates list all subsets of any set S?

f.) Provide pseudocode for all possible substrings of a single string your code could list duplicates.

9.)
a.)How many ways can you make a group of 2 out of S= {y, b, g}. Show the groups.

b.) Provide pseudocode to list all sets of 2 given S.

10.) Analysis

```
for(int i=1 to i=n)
    for(j=1 to j=n){
        j++;
        i++;
    }//for j
//for i
```

What is the runtime of the code above?

11.) Provide pseudocode to find the smallest number in a array of ints. What is the runtime of your code?

12.) Provide pseudocode to finding the sum of an array of ints. What is the runtime of your code?