ECS 122a

Due: Friday April 17, 2020 Midnight

Problem Set 2

Solving Recurrence with Substitution

1.) 4.3-1

2.) 4.3-2

3.) 4.3-7

Solving Recurrence with recursion tree method

4.) 4.4-3

5.) 4.4-4

Master's Theorem

6.) 4.5-1

7.) 4.5-4,

8.) 4-2 a & b,

Algorithms, Analysis and solving Recurrences

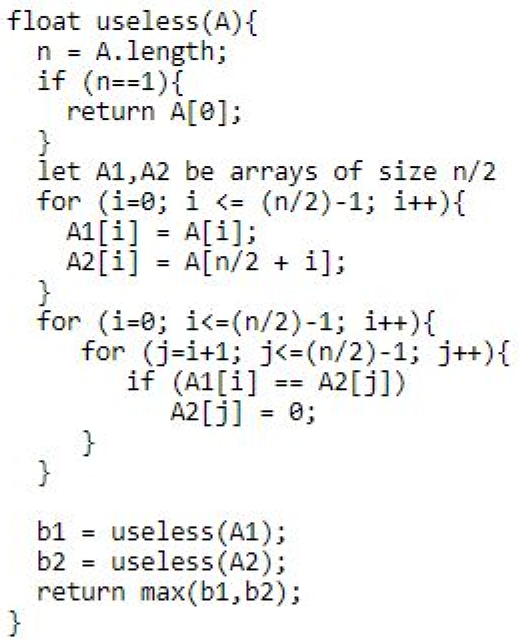
10.) (a) Show how to Merge (not Mergesort) three ordered lists each of size n/3, with a total of at most $\frac{5}{3}n$ compare operations.

(b) Let's examine a new Merge sort algorithm, where parameter A in (A,n) is a list of numbers to be sorted and n is the total length of list A.

```
Procedure mergeSort (A,n)::
    if
        n=1 then return A
    Else
        divide A into three equal size lists B, C, D,
        B := mergesort(B,n/3)
        C :=mergesort(C,n/3)
        D:=mergesort(D,n/3)
    return merge(B,C, D)
    End.
```

Set up and solve a recurrence relation to analyze the worst-case number of operations of compares that this version does.

11.)Review 2



```
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 is the asymptotic upper bound of the code above?