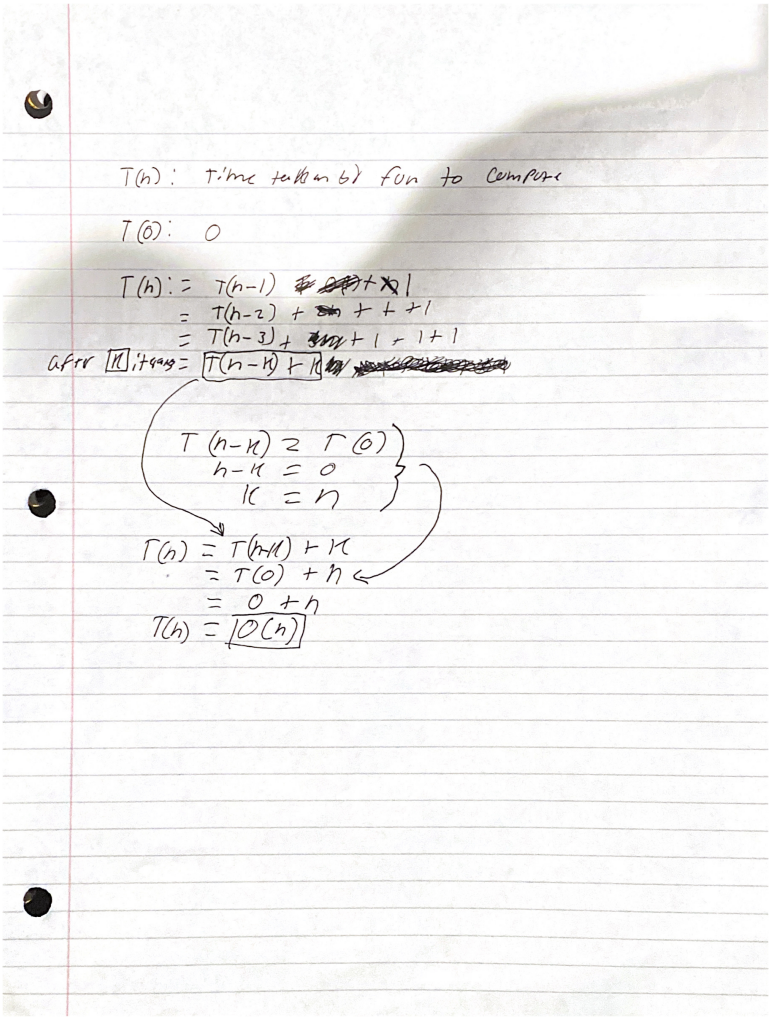
/*
Jason Wilson
Thursday, February 27th 2020
CMPT435L 111 20S
*/

Assignment 4

1. Analyze the following code and provide a "Big-O" estimate of its running time in terms of n. Explain your analysis.

```
function fun(int n)
{
    if (n == 0)
            return 0;
    else
            return n + fun(n-1);
}
```

Note: Credit will not be given only for answers - show all your work:
(3 points) steps you took to get your answer.
(2 points) your answer.



2. We are given an array A[] of n numbers in an arbitrary order. Design an algorithm to find the minimum and second minimum element in A[] using at most 3/2n -2 comparisons.
   (i) describe the idea behind your algorithm in English (3 points);

1. Set the base case of the algorithm to be if the array index difference j - I equals 0 then you have found min and min2nd to be the same because the search size is 1
2. The second base case is if j - I is equal to 1
3. Compare the two elements in the array and find min and min2nd
4. The general case is splitting the array into two equal halves and calling the function recursively until the min and min2nd of the left and right half are returned
5. compare the two halves and return min and mind2nd

   (ii) provide pseudocode (5 points);

```
dcfindmin2ndmin( int A[], int i, int j)
        Min2ndMin pair
        if (j - i = 0)
                pair.min = i
                pair.min2nd = j;
        end if
        else if (j - i = 1)
                if (A[i] > A[j])
                        pair.min = j
                        pair.min2nd = i
                end if
                else
                        pair.min = i
                        pair.min2nd = j
                end else
        end else if
        else
                int mid = (i + j) / 2, int startL = i, int endL = mid, int startR = mid + 1,
                int endR = j, Min2ndMin pairL, Min2ndMin pairR
                pairL = dcfindmin2ndmin(A, startL, endL)
                pairR = dcfindmin2ndmin(A, startR, endR)
                if A[pairL.min] > A[pairR.min])
```

```
            pair.min = pairR.min
            pair.min2nd = min(A, pairL.min, pairR.min2nd)
        end if
        else
            pair.min = pairL.min
            pair.min2nd = min(A, pairR.min, pairL.min2nd)
        end else
    end else
OUTPUT pair // contains pair.min and pair.min2nd
```

(iii) analyze the number of comparisons used in your algorithm (2 points).

$$C(n) = 2C(\frac{n}{2}) + 2$$

$$C(n) = 2C(\frac{n}{4}) + 2$$

$$C(n) = 2C(\frac{n}{8}) + 2$$

$$C(n) = 2C(\frac{n}{2}) + 2$$

$$C(n) = 2^2C(\frac{n}{2}) + 2^2 + 2$$

$$C(n) = 2^3C(\frac{n}{2}) + 2^3 + 2^2 + 2$$

$$C(n) = 2^kC(\frac{n}{2^k}) + 2^k + 2^{k-1} + 2^{k-2} + \ldots + 2^2$$

$$C(\frac{n}{2^k}) = C(2)$$

$$\frac{n}{2^k} = 2$$

$$C(n) = \frac{3}{2}n - 2$$

Note: Full credit (10 points) will be awarded for an algorithm that uses at most 3/2n -2 comparisons. Algorithms that make more comparisons will be scored out of 3 points.

3. Using the master theorem discussed in class, find a tight bound for the solution of the following recurrence equation (3 points each).

a. T(n) = 2T(n/2) + n^3

A = 2 b = 2 d = 3

$$3 > \log_2 2$$

$$O(n^3)$$

b. T(n) = T(9n/10) + n

A = 1 b = 10 d = 1

$$3 > \log_{10} 1$$

$$O(n^1) = O(n)$$

c. T(n) = 16T(n/4) + n^2

a = 16 b = 4 d = 2

$$2 = \log_4 16$$

$$n^2 \log n$$

d. T(n) = 7T(n/3) + n^2

A = 7 b = 3 d = 2

$$2 > \log_3 7$$

$$O(n^2)$$

e. T(n) = 2T(n/4) + sqrt(n)

A = 2 b = 4 d = .5

$$.5 = \log_4 2$$

$$O(n^{.5} \log n)$$

Section 2: Java Implementation

4. Implement problem 2 in Java (30 points).
   Note:

   Find a file called Problem2.java in assignment 4 folder.

   Complete the method of dcfindmin2ndmin ().

Test your method in the main method provided following the comments.

Full credit (30 points) will be awarded for an algorithm that uses at most 3/2n -2 comparisons. Programs that make more comparisons will be scored out of 5 points.

TURN-IN CHECKLIST:

1. **Answers to Section 1 (.doc/.txt), and to Section 2 (all your source Code (.java files)). Remember to include your name, the date, and the course number in comments near the beginning of your code/report.**

2. **Create a folder and name it 'FirstName_LastName_assignment_4'. In the newly created folder copy and paste your files (.doc/.txt/.java files). Then compress the folder, and submit to iLearn.**