Algorithms -- COMP.4040 Honor Statement
(Courtesy of Prof. Tom Costello and Karen Daniels with modifications)

Must be attached to each submission

Academic achievement is ordinarily evaluated on the basis of work that a student produces independently. Infringement of this Code of Honor entails penalties ranging from reprimand to suspension, dismissal or expulsion from the University.

Your name on any exercise is regarded as assurance and certification that what you are submitting for that exercise is the result of your own thoughts and study. Where collaboration is authorized, you should state very clearly which parts of any assignment were performed with collaboration and name your collaborators.

In writing examinations and quizzes, you are expected and required to respond entirely on the basis of your own memory and capacity, without any assistance whatsoever except such as what is specifically authorized by the instructor.

I certify that the work submitted with this assignment is mine and was generated in a manner consistent with this document, the course academic policy on the course website on Blackboard, and the UMass Lowell academic code.

Date: 02/06/2019

Name (please print): DangNhi Ngoc Ngo

Signature:

COMP.4040 **HW1** Sp19 - Dr. Lin

Due Date: Feb. 06, 2019 (W), BEFORE the lecture starts

This assignment covers textbook Chapter1~2. A paper version must be submitted. Please keep a copy of your solution for yourself.

1. **Compare Functions**: (20 points) Textbook, Exercise 1.2-3 (page 14) Justify your answer. (Hint: You may write a program, draw a plot, or proof)

2. **Pseudocode and Loop Invariant**: (20 points) textbook, Exercise2.1-3, p22, Linear Search.

3. **Sorting Algorithms**: (20 points) Using textbook Figure 2.2 and Figure 2.4 as models to illustrate the operations of Insertion_Sort and Merge_Sort on the array A = <31, 41, 59, 26, 41, 58>

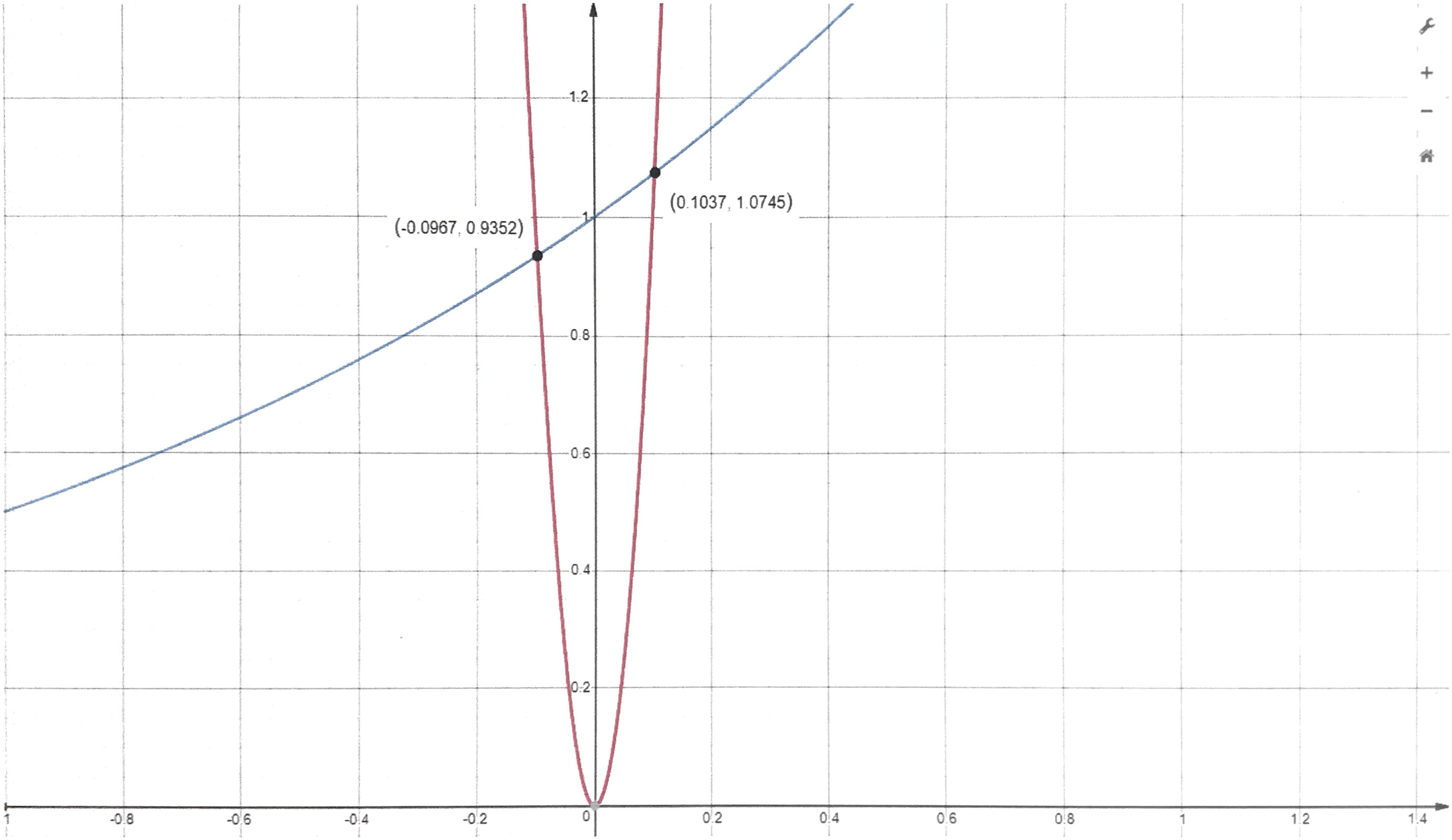
4. **Analysis**: (20 points) There is a mystery function called Mystery(n) and the pseudocode of the algorithm is shown as below. Please analyze the worst-case asymptotic execution time of this algorithm using the method we learn in the class. Express the execution time as a function of the input value *n*. Assume that $n = 3^k$ for some positive integer $k \geq 1$. Justify your answer by drawing a recursion tree to help your calculation as we have learned in the class. Appendix A may help your calculation if needed.

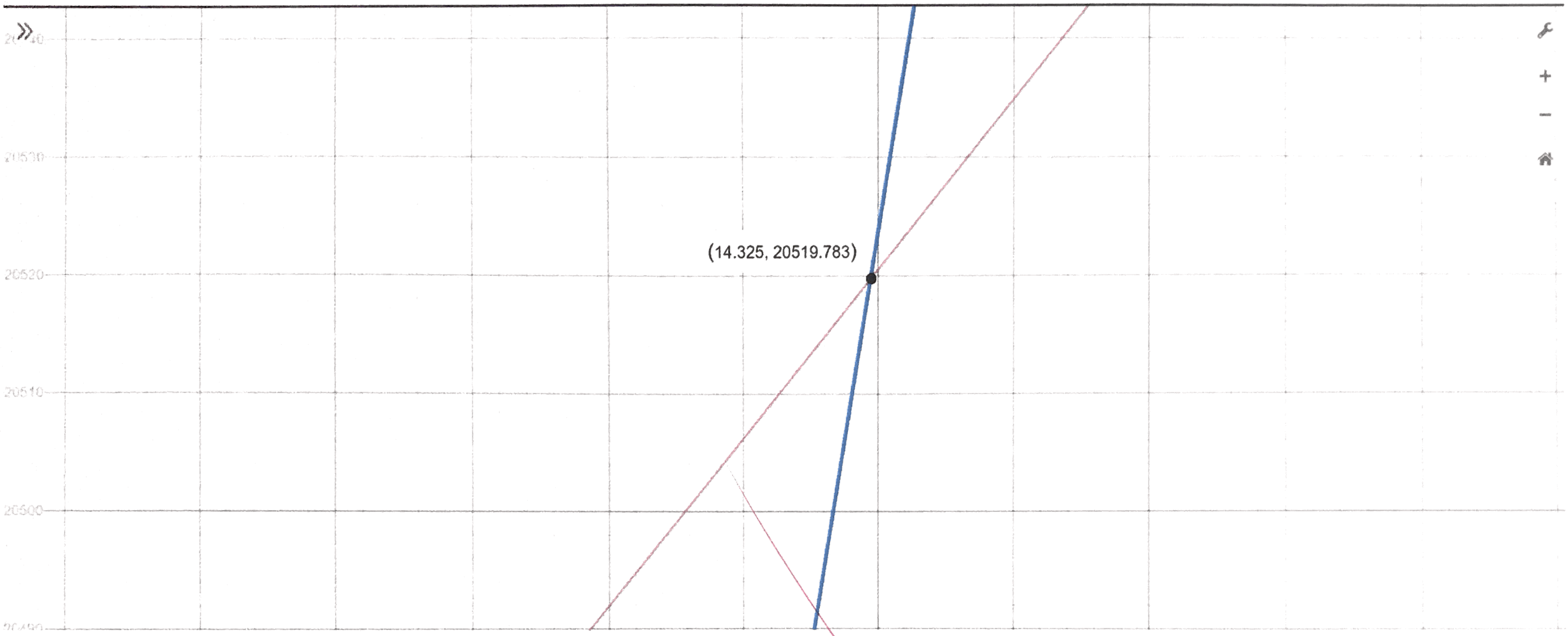
```
Mystery($n$)
    if $n \le 1$
        return 1
    for $i$=1 to 5
        for $j$ = 1 to $n$
            print "this is a recursive call."
    Mystery ($n/3$)
    Mystery ($n/3$)
    Mystery ($n/3$)
```

5. **Algorithm Design** (20 points)
   Input: array A contains *n* distinct numbers from 1 to n, in arbitrary order.
   Output: *number of inversions (*defined as the number of pair(i, j) of array indices with i < j and A[i] > A[j]).
   (a) (5 points) What array with elements from the set {1, 2, ..., n} has the most inversions? How many does it have?
   (b) (15 points) Create an algorithm using divide-and-conquer approach that determines the number of inversions in any permutation on n elements in Θ(nlgn) worst-case time (Hint: modify the merge sort).

(The Honor Statement is in the next page. Please sign and attach it to your homework solution as the last page.)

1/20


(-0.0967, 0.9352)
(0.1037, 1.0745)
1.2
1
0.8
0.6
0.4
0.2
0
-1
-0.8
-0.6
-0.4
-0.2
0.2
0.4
0.6
0.8
1
1.2
1.4



There are 3 intersections: $(-0.0967, 0.9352)$, $(0.1037, 1.0745)$, $(14.325, 20519.783)$

The smallest value of n is 15, because $100x^2 < 2^x$ and $n > 0$.

Check: $100 \times 14^2 = 19600 > 2^{14}$ $\Rightarrow$ False

$100 \times 15^2 = 22500 < 2^{15}$ $\Rightarrow$ True

Pseudocode and Loop Invariant

Exercise 2.1-3 Linear Search

20

```
Linear_Search (A, v)
    for i = 1 to A.length
        if A[i] == v
            return i
    return NIL
```

Loop invariant: At the start of each iteration of for loop $A[i] \neq v$

Initialization : The function shows true before first iteration

Maintenance : The loop invariant holds true for every iteration

If a match is found, the function will return

Termination : The function will either return an index or NIL when the loop ends

3/ Sorting Algorithms

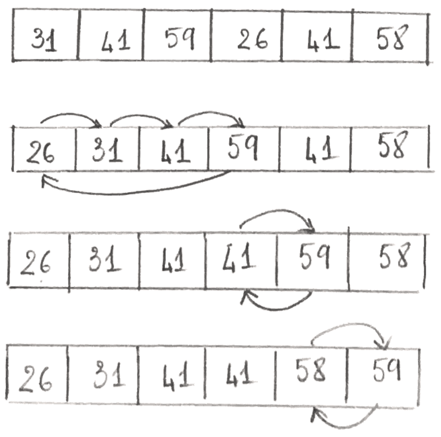
15

Array A = < 31, 41, 59, 26, 41, 58 >

* Insertion_Sort

start :

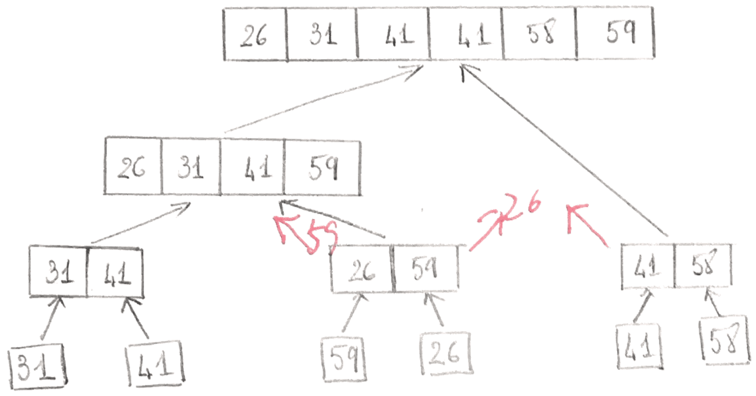
sorted :



* Merge_Sort

sorted:

start :

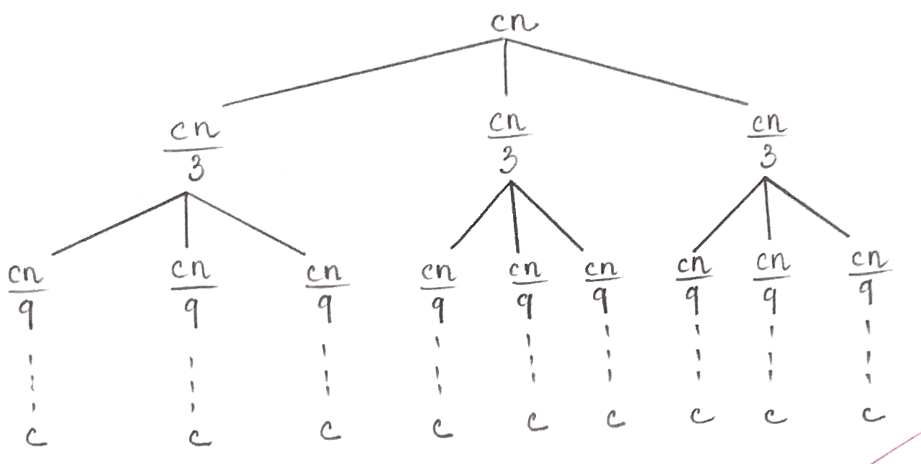


4/ Analysis

20

$$T(n) = 3T\left(\frac{n}{3}\right) + cn$$

Recursion tree:



Level	Each node in terms of level cn
#1	cn
#2	$\frac{cn}{3}$
#3	$\frac{cn}{9}$
#i	$\frac{cn}{3^{i-1}}$

$$\frac{cn}{3^{i-1}} = c \iff n = 3^{i-1}$$

$$\iff \log_3 n = i - 1$$

$$\Rightarrow i = \log_3 n + 1$$

$$T(n) = cn(\log_3 n + 1)$$

$$= cn \log_3 n + cn$$

$$= \Theta(n \log_3 n)$$

20 5/ Algorithm Design

a/ Array with the elements in the reverse order $\{n, n-1, n-2, \dots, 3, 2, 1\}$ from the set $\{1, 2, \dots, n\}$ has the most inversions

Number of inversions = Sum of 1 to $(n-1)$

$$= 1 + 2 + 3 \dots + (n-1)$$

$$= \frac{n(n+1)}{2} - n$$

$$= \frac{n^2 + n - 2n}{2}$$

$$= \frac{n(n-1)}{2}$$

5/b/ Create an algorithm using divide-and-conquer approach 95/100

Number of inversions:

```
Inversion (A, p, q, r)
$n_1 = q - p + 1$
$n_2 = r - q$
Let $L[1 \dots n_1 + 1]$ and $R[1 \dots n_2 + 1]$ be new arrays
for $i = 1$ to $n_1$
    $L[i] = A[p + i - 1]$
for $j = 1$ to $n_2$
    $R[j] = A[q + j]$
$L[n_1 + 1] = \infty$
$R[n_2 + 1] = \infty$
$i = 1$
$j = 1$
numOfInversions = 0
count = FALSE
for $k = p$ to $r$
    if count = FALSE and $R[j] < L[i]$
        numOfInversions += $n_1 - i + 1$
        count = TRUE
    if $L[i] \leqslant R[j]$
        $A[k] = L[i]$
        $i = i + 1$
    else $A[k] = R[j]$
        $j = j + 1$
        count = FALSE
return numOfInversions
```