CIS 477 (Fall 2022) Disclosure Sheet

Name: Kevin Lopez

HW # 3

Yes No Did you consult with anyone on parts of this assignment, including other students, TAs, or the instructor?

Yes No Did you consult an outside source (such as an Internet forum or a book other than the course textbook) on parts of this assignment?

If you answered Yes to one or more questions, please give the details here:

By submitting this sheet through my Blackboard account, I assert that the information on this sheet is true.

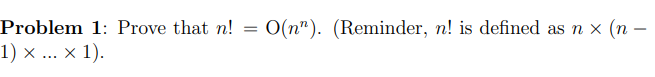
This disclosure sheet was based on one originally designed by Profs. Royer and Older.

Kevin Lopez

CIS477

10/3/22

HW2

1. 
   1. Claim: n! = O(nn)
   2. Direct Proof:
      1. For n=0, n! = 0
      2. For n=1, n!=0
      3. For n, n!= n\*(n-1)\*….\*1
      4. For n+1, (n+1)\*n\*(n-1)…\*1
      5. This can be rewritten as (n+1)\*n!
      6. This can be rewritten as (n2+n) \*(n-1)!
      7. This can be rewritten as (n3-n)
      8. We can see that the n is increasing as well as the exponent raised to it, so it is (nn-1+c)
      9. After this, We see the recurrence relation is showing O(nn)

Text

Description automatically generated

* 1. T(n) = running time of FindMin for array size n
  2. T(1) = C1
  3. T(n) = C2 + T(n/2) + T(n/2) + O(1)
  4. T(n) = 2T(n/2) + O(1)
  5. A = 2, b = 2, d = 0
  6. Log22 < 0
  7. O(n)

1. Text

   Description automatically generated
   1. Merge Sort
      1. Array1 = [6,8] , Array2 = [7,1,2] , mergedArray = []
      2. Array1 = [6,8] , Array2 = [7,2] , mergedArray = [1]
      3. Array1 = [6,8] , Array2 = [7] , mergedArray = [1,2]
      4. Array1 = [8] , Array2 = [7] , mergedArray = [1,2,6]
      5. Array1 = [8] , Array2 = [] , mergedArray = [1,2,6,7]
      6. Array1 = [] , Array2 = [] , mergedArray = [1,2,6,7,8]
   2. InsertionSort
      1. Array = [6,8,7,1,2]
      2. Array = [6,7,8,1,2]
      3. Array = [1,6,7,8,2]
      4. Array = [1,2,6,7,8]
   3. BubbleSort
      1. Array = [6,8,7,1,2]
      2. Array = [6,7,8,1,2]
      3. Array = [6,7,1,8,2]
      4. Array = [6,7,1,2,8]
      5. Array = [6,7,1,2,8]
      6. Array = [6,1,7,2,8]
      7. Array = [6,1,2,7,8]
      8. Array = [6,1,2,7,8]
      9. Array = [1,6,2,7,8]
      10. Array = [1,2,6,7,8]
   4. Quicksort
      1. Array = [6,8,7,1,2]
      2. Array = [1,2,6,8,7]
      3. Array = [1,2,6,7,8]
2. Graphical user interface

   Description automatically generated with low confidence
   1. T(n) = worst case scenario for Quicksort for array size n
   2. T(1) = C1
   3. T(n) = T(n-1) + T(n-2)+…+T(n)
   4. T(n) = (n-1) + (n-2) + … + n
   5. T(n) = ((n-1)n)/2
   6. T(n) = O(n2)
3. Text, letter

   Description automatically generated
   1. The values of these are
      1. B3=6
      2. B5=15
      3. B7=28
   2. B(n) = running time of B for binary tree size n
   3. B(1) = C1
   4. B(n) = 2B(n) + n