CIS 477 (Fall 2022) Disclosure Sheet

Name: Kevin Lopez

HW # 1

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
09/16/22
CIS477
HW1

1) For all pairs of functions below, prove (with explanation) whether f(n) is O(g(n)) or g(n) is O(f(n)) (or both). (You might find it helpful to look up the rules of logarithms).
   a) f(n)=n^2 g(n)=n^4 – n^3
      i) Proof: Prove that f(n)=O(g(n)) but g(n) is not big O of f(n)
      ii) Direct proof: using limit theorem
         (1) $\text{Lim}_{x\text{->}\infty}$(f(n)/g(n))
         (2) $\text{Lim}_{x\text{->}\infty}$(n^2/n^4-n^3)
         (3) Using L'Hopital's rule:
            (a) $\text{Lim}_{x\text{->}\infty}$(2n/4n^3 - 3n^2)
            (b) $\text{Lim}_{x\text{->}\infty}$(2/12n^2 – 6n)

iii) This goes to to 0, therefore using the Limit theorem we see that f(n) is Big O of g(n) but not the other way around.

b) f(n) = $\log_2(6n)$ g(n)=$\log_4(n)$

i) Proof: These two functions are Big O of each other

ii) Direct Proof: using limit theorem

(1) $\text{Lim}_{x\to\infty}(f(n)/g(n))$

(2) $\text{Lim}_{x\to\infty}(\log_2(6n) \ / \log_4(n))$

(3) Using L'hopital's rule:

(a) $\text{Lim}_{x\to\infty}(\log_2(6n) \ / \log_4(n))$

(b) $\text{Lim}_{x\to\infty}((1/n(\ln(2)) \ / (1/(n\ln(4)))$

(4) Simplication:

(a) $\text{Lim}_{x\to\infty}(n\ln(4))/n(\ln(2))$

(b) $\text{Lim}_{x\to\infty}(\ln(4))/(\ln(2))$

iii) This number is between 0 and infinity therefore they are Big O of each other

c) f(n) =$3\log_2 n$ g(n)= $\log_2$(n^8)

i) Proof: f(n) is Big O of g(n) but not the other way around

ii) Direct Proof: Using limit theorem

(1) $\text{Lim}_{x\to\infty}(f(n)/g(n))$

(2) $\text{Lim}_{x\to\infty}(3\log_2 n / \log_2(n\text{^}8))$

(3) $\text{Lim}_{x\to\infty}(3/n\ln(2)) / (8n^7/n^8\ln(2))$

iii) Simplification:

(1) $\text{Lim}_{x\to\infty}(3n^8\ln(2)/8n^8\ln(2))$

(2) $\text{Lim}_{x\to\infty}(3/8)$

iv) This limit is between 0 and infinity therefore f(n) = O(g(n)) and g(n) = O(f(n))

2) Consider the following algorithm:

```
function SumElement(array $A$)
    if $length(A) == 1$:
        return $A[0]$
    else:
        $A1 \leftarrow A[0 : length(A)/4]$
        $A2 \leftarrow A[length(A)/4 : 2 * length(A)/4]$
        return SumElement(A1) + SumElement(A2)
end function
```

a)

b) Write the running time of this function as a recurrence relation.

i) T(n) represents the running time of SumElement for an array length n.

ii) T(1) = Constant

iii) T(n)= Constant2 + 2T(n/4)

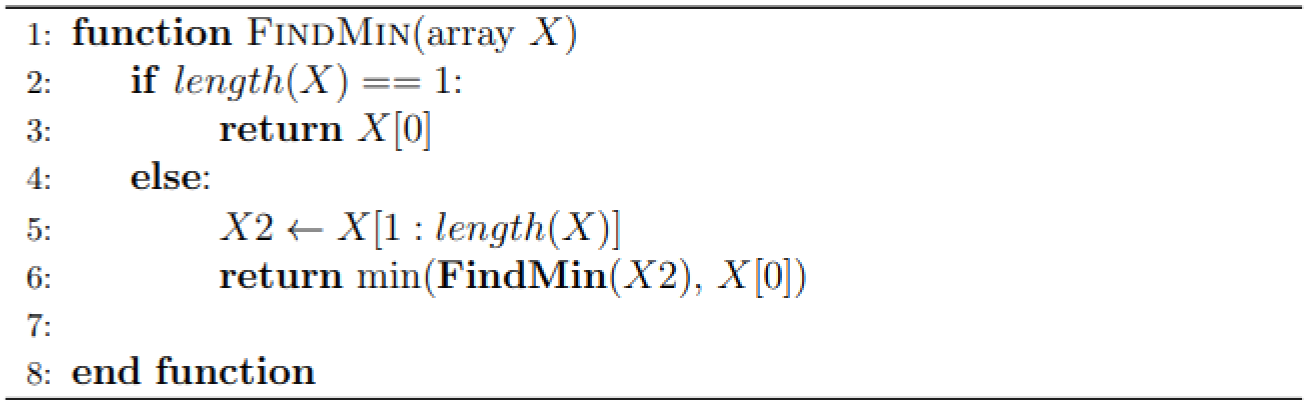
iv) T(n) = $2\log_4 n$ *Constant2 + Constant3

c) Describe the running time of this function using big-O notation.

i) O(log(n))

3) . Determine whether the following statements are true or false, and prove your answer:
   a) • If f(x) = O(g(x)), then g(x) = O(f(x)).
      i) Claim: f(x)=O(g(x)), then g(x) is not o(f(x))
      ii) Proof By Contradiction
         (1) Assume f(x)=O(g(x)) , then g(x) = o(f(x)).
         (2) Let $f(x)=x^2$ and g(x) = x
         (3) Limit theorem:
            (a) $\text{Lim}_{x\text{->}\infty}$(f(x)/g(x))
            (b) $\text{Lim}_{x\text{->}\infty}$(x^2 / x)
            (c) The number is between 0 and infinity, so according to the definition, f(x)=O(g(x) but not the other way around
      iii) Therefore, By contradiction, we have proved a case where f(x)=O(g(x)) but it is not guaranteed to be the other way around
   b) • if f(x) is not O(g(x)), then f(x) is not o(g(x)).
      i) Claim: if f(x) is not O(g(x)), then f(x) is not o(g(x))
      ii) Proof by Contradiction:
         (1) Assume f(x) is not O(g(x)), then f(x) is not o(g(x))
         (2) We can change the statement if f(x) is not O(g(x)), then g(x) is not o(g(x)) to if f(x) is O(g(x)) then f(x) is o(g(x)) using Logic equivalencies
         (3) Let f(x) = x^3 and g(x) = x^3 + 3
         (4) Using limit theorem, we see that they are big O of each other but f(x) is not o(g(x))
      iii) Therefore by Contradiction, The claim is True

4) Analyze the running time of the following algorithm (assume arrays are 0-indexed)



```
1: function FINDMIN(array $X$)
2:     if $length(X) == 1$:
3:         return $X[0]$
4:     else:
5:         $X2 \leftarrow X[1 : length(X)]$
6:         return min(FindMin($X2$), $X[0]$)
7:
8: end function
```

   a)
   b) O(n) because it is doing a single min function at the end after building the stack linearly.
   c) Cost of lines
      i) 1
      ii) 1
      iii) C1
      iv) 0
      v) n-1
      vi) 2C2=T(n-2)
   d) T(n) = 1 + 1 + (n-1) + T(n-1)
   e) T(n) = 1 + n + T(n-1)

    f) This is O(n)

5) Suppose that Alice is asked to prove that the running time of an algorithm is O(n), where n is the size of the input to that function. To prove this, she programs the algorithm on her computer, runs it for many different sizes of input, and makes a plot showing the running time of the code for the various values of n. When she does this, it is apparent that the shape of the plot is linear. Thus, she concludes that the running time of the function is O(n). In your own words, explain what is wrong with Alice's 'proof'.
    a) Alice's proof is wrong for two main reasons. One reason is because she is doing it on the same machine so she is not seeing how it is running on different platforms. The second reason is because she is creating specific test cases that can make it seem linear, but with other ones they may have different run times.