CSCI 3104 PS1a

Jonathan Phouminh

TOTAL POINTS

8 / 10

QUESTION 1

1 3 / 3

✓ **+ 3 pts Correct**

+ 2 pts Something wrong about maintenance step. Please check solution, you need to clarify that LI is true before/after each iteration of loop.

+ 1 pts only one component is correct

+ 1 pts Generally in right direction, but incorrect.

+ 1 pts No one-sentence description for each one.

+ 2 pts Your Maintenance section is not correct

+ 2 pts Termination part is not clear

+ 0 pts Not try or totally incorrect

+ 0 pts Click here to replace this description.

QUESTION 2

6 pts

2.1 2 / 2

✓ **+ 2 pts Correct**

+ 2 pts See the solution to phrase the LI better. Specifically give a context to 'i'. For ex - you can start with "At the beginning of ith iteration..".

+ 1.5 pts In the right direction but a Loop Invariant is stated differently. See solutions.

+ 0 pts Not attempted

+ 0 pts (Non-scored rubric item) - Good work on attempting the complete proof. But you were good when you stated the Loop Invariant. We are not looking at that portion of your solution now.

+ 0 pts Incorrect answer please check the solutions provided.

+ 2 pts Please also mention that at the start of the ith iteration ret has the maximum the subarray A[0, 1, ..., i-1]

+ 2 pts Please mention when (start or end of the iteration) does the variable ret contains the greatest element and in which subarray.

+ 2 pts There is no variable max, variable ret stores the maximum value.

+ 1 pts The Loop invariant representation is not right. Please check the solution.

+ 2 pts See the solution to see how Loop Invariant's are written

+ 1 pts ret holds the maximum value and not it's index.

2.2 0 / 2

+ 2 pts Correct

✓ **+ 0 pts Incorrect**

+ 2 pts Technically correct, but does not define loop invariant precisely

+ 0.5 pts The loop invariant should reflect the logic through which the algorithm works. Your answer is technically true, but it does not serve as a real loop invariant in terms of the solution to the problem.

+ 0.5 pts Partially correct. ret = -1 condition not specified correctly when n is not in the array.

+ 0.5 pts Partially correct. index value of n condition not specified correctly

💬 ret is equal to the index of n if the value has been found. ret = i, is incorrect, as the value of i changes through the loop iterations

2.3 2 / 2

✓ **+ 2 pts Correct**

+ 0 pts Not attempted

+ 1 pts Need more explanation

+ 2 pts Phrase it better. Please see the solution provided.

+ 0 pts Incorrect answer. Please read and understand the concept again

+ **0 pts** variable 'sum' is the sum of A[0....i-1], not A[0....i], before the start of each iteration. Please read again and understand the subarray concept.

+ **0 pts** Can be written more clearly

+ **0 pts** The sum is equal to the sum of the elements in the subarray A[0:i-1], not just the elements in the subarray.

+ **0 pts** Please write maintenance step properly.

+ **1 pts** Answer seems to be in the right direction, but it is not clear and enough.

QUESTION 3

3 1 / 1

+ **0 pts** Incomplete or Incorrect Base case and Inductive Hypothesis, or Empty Solution submitted. Please check the solution for reference.

✓ **+ 1 pts Correct and Great Work !!!**

+ **0.5 pts** Missing or Incorrect Base Case. Please Check Solution File.

+ **0.5 pts** Missing or Incorrect Inductive Hypothesis. Please Check Solution File for reference.

+ **1 pts** In the right direction. But please compare your solution with the actual solution file so that you do perfect the next time.

+ **1 pts** In the right direction. Missing what LHS and RHS are, but proved both the sides to be equal. Please check solution file for your reference and do better the next time.

+ **1 pts** In the right direction. Please check the corrected question and solution file for reference so that you do perfect the next time.

+ **0.5 pts** In the right direction. But please compare your solution with the actual solution file so that you do perfect the next time.

+ **0.5 pts** In the right direction, but incorrect question assumed and hence tried to prove the wrong statement.

Name: Jonathan Phouminh

ID: 106054641

CSCI 3104, Algorithms **Profs. Hoenigman & Agrawal**
Problem Set 1a (10 points) **Fall 2019, CU-Boulder**

Advice 1: For every problem in this class, you must justify your answer: show how you arrived at it and why it is correct. If there are assumptions you need to make along the way, state those clearly.

Advice 2: Verbal reasoning is typically insufficient for full credit. Instead, write a logical argument, in the style of a mathematical proof.

Instructions for submitting your solution:

- The solutions **should be typed** and we cannot accept hand-written solutions. Here's a short intro to Latex.
- You should submit your work through **Gradescope** only.
- If you don't have an account on it, sign up for one using your CU email. You should have gotten an email to sign up. If your name based CU email doesn't work, try the identikey@colorado.edu version.
- Gradescope will only accept **.pdf** files (except for code files that should be submitted separately on Gradescope if a problem set has them) and **try to fit your work in the box provided**.
- You cannot submit a pdf which has less pages than what we provided you as Gradescope won't allow it.

Name: Jonathan Phouminh

ID: 106054641

CSCI 3104, Algorithms **Profs. Hoenigman & Agrawal**
Problem Set 1a (10 points) **Fall 2019, CU-Boulder**

1. *(3 pts) What are the three components of a loop invariant proof? Write a one-sentence description for each one.*

a) Initialization: LI is true prior to execution of the loop

b) Maintenance : Suppose LI holds true before the beginning of the loop, it remains true after the body executes.

c) Termination: The LI is true after the loop terminates

2. *(6 pts total) Identify the loop invariant in the following algorithms.*

(a)
```
FindMaxElement(A) : //suppose array A is not empty
        ret = A[0]
        for i = 1 to length(A)-1 {
            if A[i] > ret{
                ret = A[i]
        }}
        return ret
```

The loop invariant variable is ret. Since the sub array has the only element that has been seen initially, it must be the max element. Thus before every iteration ret will always have the max value from A[0:i-1]

(b)
```
FindElement(A, n) : //suppose no duplicates in array A and array A is not empty
    ret = -1 //index -1 implies the element haven't been found yet
    for i = 0 to length(A)-1 {
        if A[i] == n{
            ret = i
    }}
    return ret
```

The loop invariant variable is ret has multiple properties such that ret will always be equal to -1 or i before the start of every iteration.

(c)
```
SumArray(A) : //suppose array A is not empty
    sum = 0
    for i = 0 to length(A)-1 {
        sum += A[i]
    }
    return sum
```

The loop invariant variable is sum. Since since the array has not been traversed nothing has been added therefore sum = 0. Therefore sum will always have the summation of elements after every iteration from A[0:i-1]

Name: Jonathan Phouminh

ID: 106054641

CSCI 3104, Algorithms **Profs. Hoenigman & Agrawal**

Problem Set 1a (10 points) **Fall 2019, CU-Boulder**

3. *(1 pt) If r is a real number not equal to 1, then for every $n \geq 0$,*

$$\sum_{i=0}^{n} r^i = \frac{(1 - r^{n+1})}{(1 - r)}.$$

Provide the first two steps of a proof by induction i.e. base case and the inductive hypothesis. You will be asked to complete this proof later in ***PS1b****.*

Base Case, let n = 0

$$\sum_{i=0}^{0} r^i = \frac{(1 - r^{0+1})}{(1 - r)} 1 = \frac{(1 - r)}{(1 - r)} 1 = 1$$

Inductive Hypothesis
Assume,

$$\sum_{i=0}^{k} r^i = \frac{(1 - r^{k+1})}{(1 - r)}.$$

show n = $k + 1$ holds

$$\sum_{i=0}^{k+1} r^i = \frac{(1 - r^{(k+1)+1})}{(1 - r)}.$$