Lance Bunch

7/19/2021

CSE 130: Algorithm Design

Brother Bailey

**Pseudocode**

GET classID

n <- len(classID) + 1 # A

sum <- (n \* (n + 1)) / 2 # B

sumArr <- 0 # C

FOR i IN range(0, n – 1) # D

sumArr <- sumArr + classID[i] # E

missingStudent <- sum – sumArr # F

PUT (missingStudent)

**Program Trace**

**Test Case: classID[1, 4, 3, 5]**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **n** | **sum** | **sumArr** | **i** | **missingStudent** |
| **A** | 5 | - | - | - | - |
| **B** | 5 | 15 | - | - | - |
| **C** | 5 | 15 | 0 | - | - |
| **D** | 5 | 15 | 0 | 0 | - |
| **E** | 5 | 15 | 1 | 0 | - |
| **D** | 5 | 15 | 1 | 1 | - |
| **E** | 5 | 15 | 5 | 1 | - |
| **D** | 5 | 15 | 5 | 2 | - |
| **E** | 5 | 15 | 8 | 2 | - |
| **D** | 5 | 15 | 8 | 3 | - |
| **E** | 5 | 15 | 13 | 3 | - |
| **D** | 5 | 15 | 13 | 4 | - |
| **F** | 5 | 15 | 13 | - | 2 |

**Algorithmic Efficiency**

This algorithm is extremely efficient as it caps out at *O(n)* efficiency. We know this because there is one FOR loop, and this loop runs through each element of an array that is given by the user. If the array has 15 elements it will run through all 15 elements. If the array is 1,000 elements it will run through all 1,000 elements.