# Riphah School of Computing and Innovation (RSCI), Lahore

****

**Data Structures and Algorithms**

## 3rd Semester

## Assignment 1

**Submitted to:** Ms. Atika Islam

**Student Id:**

58155

**Student Name:**

Mahnoor Islam

BSCS-3B

04/11/2024

#include <iostream>

using namespace std;

double findMedianSortedArrays(int class1[], int size1, int class2[], int size2) {

int totalSize = size1 + size2;

int medianIndex1 = (totalSize - 1) / 2;

int medianIndex2 = totalSize / 2;

int i = 0, j = 0, count = 0;

int current = 0, previous = 0;

while (count <= medianIndex2) {

previous = current;

if (i < size1 && (j >= size2 || class1[i] <= class2[j])) {

current = class1[i];

i++;

} else {

current = class2[j];

j++;

}

count++;

}

if (totalSize % 2 == 1) {

return current;

}

return (previous + current) / 2.0;

}

int main() {

int class1[] = {55, 70, 85};

int class2[] = {60, 75, 90, 100};

int size1 = sizeof(class1) / sizeof(class1[0]);

int size2 = sizeof(class2) / sizeof(class2[0]);

double median = findMedianSortedArrays(class1, size1, class2, size2);

cout << "Median = " << median << endl;

    return 0;

}

**DRY RUN:**

**Main():**

**Step 1:**

**Memory:** Class1 = {55,70,85}

**Step 2:**

**Memory:** Class1 = {55,70,85}, Class2 = {60, 75, 90, 100}

**Step 3:**

**Memory:** Class1 = {55,70,85}, Class2 = {60, 75, 90, 100}, size1(class 1 size) = 3

**Step 4:**

**Memory:** Class1 = {55,70,85}, Class2 = {60, 75, 90, 100}, size1(class 1 size), = 3,size2(Class 2 size) = 4

**Step 5:**

**Memory:** Class1 = {55,70,85}, Class2 = {60, 75, 90, 100}, size1(class 1 size), = 3,size2(Class 2 size) = 4, median = findMedianSortedArrays(class1, size1, class2, size2);

**Logic :** findMedianSortedArrays({55,70,85},3 , {60, 75, 90, 100}, 4);

**Funtion Memory:** Array1 = {55,70,85}, Array1 = {60, 75, 90, 100}, size1(class 1 size), = 3,size2(Class 2 size) = 4

Function Line 1:

**Funtion Memory:** Array1 = {55,70,85}, Array1 = {60, 75, 90, 100}, size1(class 1 size), = 3,size2(Class 2 size) = 4,

totalSize = 3+4 = 7

Function Line 2:

**Funtion Memory:** Array1 = {55,70,85}, Array1 = {60, 75, 90, 100}, size1(class 1 size), = 3,size2(Class 2 size) = 4,

totalSize = 7

medianIndex1 = (7 - 1) / 2 = 3

Function Line 3:

**Funtion Memory:** Array1 = {55,70,85}, Array1 = {60, 75, 90, 100}, size1(class 1 size), = 3,size2(Class 2 size) = 4,

totalSize = 7

medianIndex1 = 3

medianIndex2 = 7 / 2 = 3

Function Line 4,5:

**Funtion Memory:** Array1 = {55,70,85}, Array1 = {60, 75, 90, 100}, size1(class 1 size), = 3,size2(Class 2 size) = 4,

totalSize = 7

medianIndex1 = 3

medianIndex2 = 3

i=0,j=0,count=0,current=0,previous=0

Function Line 6:

**Funtion Memory:** Array1 = {55,70,85}, Array1 = {60, 75, 90, 100}, size1(class 1 size), = 3,size2(Class 2 size) = 4,

totalSize = 7

medianIndex1 = 3

medianIndex2 = 3

i=0,j=0,count=0,current=0,previous=0

**LOOP:**

Count (0)<= medianIndex2(3) -> True

**Iteration 1:**

* Previous = current (0)
* i(0)<size1(3) [True] AND (j(0)>=size2(3) [False] OR Array1[0]<=Array2[0] [True]) -> True

{

Current = Array1[i(0)] = 55

i = 0+1 = 1

}

* count = 0+1 = 1

**Iteration 2:** Count (1)<= medianIndex2(3) -> True

* Previous = current (55)
* i(1)<size1(3) [True] AND (j(0)>=size2(3) [False] OR Array1[1]<=Array2[0] [False]) -> False

{

Current = Array2[j(0)] = 60

j = 0+1 = 1

}

* count = 1+1 = 2

**Iteration 3:** Count (2)<= medianIndex2(3) -> True

* Previous = current (60)
* i(1)<size1(3) [True] AND (j(1)>=size2(3) [False] OR Array1[1]<=Array2[1] [True]) -> True

{

Current = Array1[i(1)] = 70

i = 1+1 = 2

}

* count = 2+1 = 3

**Iteration 4:** Count (2)<= medianIndex2(3) -> True

* Previous = current (70)
* i(2)<size1(3) [True] AND (j(1)>=size2(3) [False] OR Array1[2]<=Array2[1] [False]) -> False

{

Current = Array1[j(1)] = 75

j = 1+1 = 2

}

* count = 3+1 = 4

**Iteration 5 won’t happen:** Count (4)<= medianIndex2(3) -> False -> Break

Function if else condition:

totalSize(7)%2==1->True (return current(75)

else return (previous(70)+current(75))/2 {in case of even size}

**Step 6:**

**Main Memory:** Class1 = {55,70,85}, Class2 = {60, 75, 90, 100}, size1(class 1 size), = 3,size2(Class 2 size) = 4, median = 75(returned)

**Step 6:**

**Main Memory:** Class1 = {55,70,85}, Class2 = {60, 75, 90, 100}, size1(class 1 size), = 3,size2(Class 2 size) = 4, median = 75(returned)

**Console:** Output: Median = 75