





Practical Day 1 - Answer3/src/maxsumsubarray.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

maxdiffheights.java maxsumsubarray.java maxdiffheights.java firstnonrepeating.java subarray.java productofelements.java

Project Explorer

src
 (default package)
 maxsumsubarray.java
 module-info.java
JRE System Library [J2SE-1.8]
Answer4
src
 (default package)
 maxdiffheights.java
 module-info.java
JRE System Library [J2SE-1.8/Foundation 1.8]
Answer5
src
 (default package)
 firstnonrepeating.java
 module-info.java
JRE System Library [J2SE-1.8/Foundation 1.8]
Answer6
src
 (default package)
 maxdiffheights.java
 module-info.java
JRE System Library [J2SE-1.8/Foundation 1.8]

maxdiffheights.java maxsumsubarray.java maxdiffheights.java firstnonrepeating.java subarray.java productofelements.java

```
1 import java.util.*;
2
3 public class maxsumsubarray {
4     static int maxSubarraySumCircular(int[] nums) {
5         int n = nums.length;
6
7         // Case 1: Maximum sum subarray without circular wrapping
8         int maxNormal = kadane(nums);
9
10        // Case 2: Maximum sum subarray with circular wrapping
11        int totalSum = 0;
12        for (int num : nums) {
13            totalSum += num;
14        }
15
16        // Invert the signs of the array elements and find the minimum sum subarray
17        int[] invertedNums = new int[n];
18        for (int i = 0; i < n; i++) {
19            invertedNums[i] = -nums[i];
20        }
21
22        int maxCircular = totalSum - kadane(invertedNums);
23
24        // Return the maximum of the two cases
25        return Math.max(maxNormal, maxCircular);
26    }
27
28    // Kadane's algorithm to find the maximum sum subarray
29    static int kadane(int[] nums) {
30        int maxSum = nums[0];
31        int currentSum = nums[0];
32
33        for (int i = 1; i < nums.length; i++) {
34            currentSum = Math.max(nums[i], currentSum + nums[i]);
35            maxSum = Math.max(maxSum, currentSum);
36        }
37
38        return maxSum;
39    }
40 }
```

Problems Javadoc Declaration Search Console
terminated: maxdiffheights Java Application C:\Users\legend\p2\proj\plugins\org.eclipse.jdt.launcher.hotspot.jre.full.win32-x86_64-21.0.1\j2se-1.8\bin\java.exe (10-Jan-2024 9:24:42 pm - 9:24:44 pm) [pid: 11116]

Output for arr1: 5
Output for arr2: 0

6°C
Fog

Search

Writable Smart Insert 1:1:0

ENG IN

Practical Day 1 - Answer4/src/maxDifference.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

- src
 - (default package)
 - maxSubarray.java
 - module-info.java
- JRE System Library [J2SE-1.5]
- Answer4
 - src
 - (default package)
 - maxDifference.java
 - module-info.java
 - JRE System Library [CDC 1.0/Foundation 1.0]
- Answer5
 - src
 - (default package)
 - firstNonRepeating.java
 - module-info.java
 - JRE System Library [JavaSE-1.7]
- Answer6
 - src
 - (default package)
 - maxDiffHeights.java
 - module-info.java
 - JRE System Library [CDC 1.0/Foundation 1.0]

maxDiffHeights.java maxSubarray.java maxDifference.java firstNonRepeating.java pairs.java productOfElements.java

```
1 import java.util.*;
2
3 public class maxDifference {
4
5     static int maxDifference(int[] nums) {
6         if (nums == null || nums.length < 2) {
7             // Not enough elements in the array
8             System.out.println("Array should have at least two elements");
9             return -1;
10        }
11
12        int minElement = nums[0];
13        int maxDifference = nums[1] - nums[0];
14
15        for (int i = 1; i < nums.length; i++) {
16            int currentDifference = nums[i] - minElement;
17
18            // Update maxDifference if a greater difference is found
19            if (currentDifference > maxDifference) {
20                maxDifference = currentDifference;
21            }
22
23            // Update minElement if a smaller element is found
24            if (nums[i] < minElement) {
25                minElement = nums[i];
26            }
27        }
28
29        return maxDifference;
30    }
31
32    public static void main(String[] args) {
33        int[] nums = {2, 7, 9, 5, 1, 3, 5};
34        int result = maxDifference(nums);
35
36        if (result != -1) {
37            System.out.println("The maximum difference is " + result + ". The pair is (" + nums[0] + ", " + result + ")");
38        }
39    }
40}
```

Problems Javadoc Declaration Search Console

terminated: maxDiffHeights [Java Application] C:\Users\kgand\p2\pooa\plugins\org.eclipse.jst.j2ee.openjdk.hotspot.jre.full.win32.x86_64.21.0.1.v20231028-0937\jre\bin\java.exe [10-Jan-2024, 9:24]

Output for arr1: 5

Output for arr2: 8



