

## 1) Stock buy and sell

Code:

```
public class StockBuySell {  
    public static void main(String[] args) {  
        int[] prices = {7, 1, 5, 3, 6, 4};  
        System.out.println("Maximum profit: " + maxProfit(prices));  
    }  
  
    public static int maxProfit(int[] prices) {  
        int minPrice = Integer.MAX_VALUE;  
        int maxProfit = 0;  
  
        for (int i = 0; i < prices.length; i++) {  
            if (prices[i] < minPrice) {  
                minPrice = prices[i];  
            } else if (prices[i] - minPrice > maxProfit) {  
                maxProfit = prices[i] - minPrice;  
            }  
        }  
  
        return maxProfit;  
    }  
}
```

```
C:\Windows\System32\cmd.e  ×  +  ∨
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\OneDrive\Desktop\c4>javac StockBuySell.java

C:\Users\ASUS\OneDrive\Desktop\c4>java StockBuySell
Maximum profit: 5

C:\Users\ASUS\OneDrive\Desktop\c4>
```

Time complexity:  $O(n)$

2) Coin change (Count ways)

Code:

```
public class CoinChange {
    public static void main(String[] args) {
        int[] coins = {1, 2, 3};
        int amount = 4;
        System.out.println("Number of ways to make change: " +
countWays(coins, amount));
    }

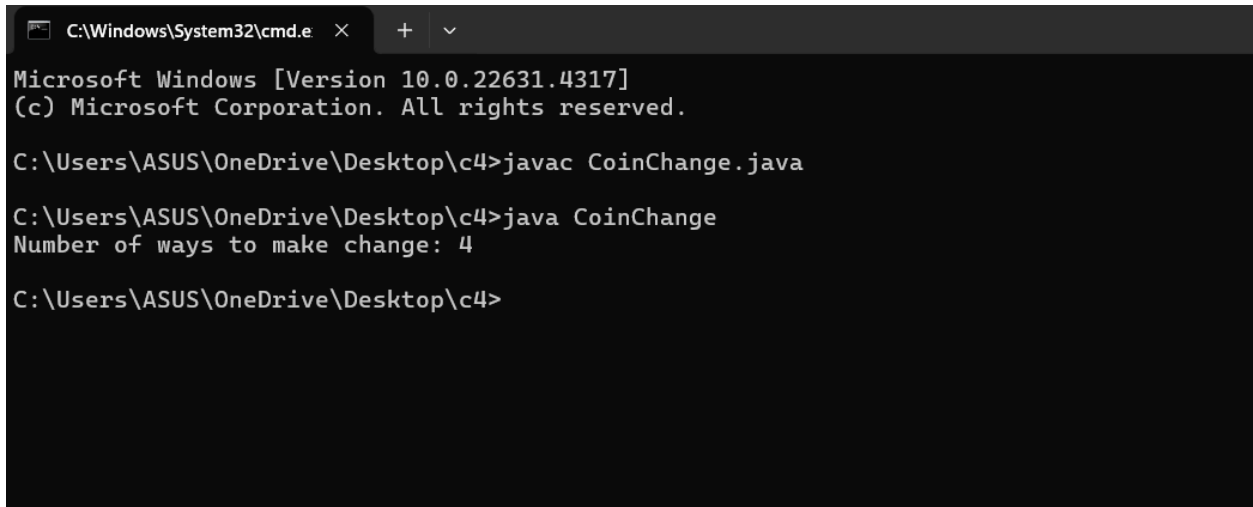
    public static int countWays(int[] coins, int amount) {
        int[] dp = new int[amount + 1];
        dp[0] = 1;
        for (int coin : coins) {
            for (int i = coin; i <= amount; i++) {
                dp[i] += dp[i - coin];
            }
        }
    }
}
```

```

    }
}

return dp[amount];
}
}

```



A screenshot of a Windows command prompt window. The title bar shows 'C:\Windows\System32\cmd.e' with a close button and a dropdown arrow. The window content displays the following text:

```

Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\OneDrive\Desktop\c4>javac CoinChange.java

C:\Users\ASUS\OneDrive\Desktop\c4>java CoinChange
Number of ways to make change: 4

C:\Users\ASUS\OneDrive\Desktop\c4>

```

Time complexity:  $O(n \cdot m)$

### 3) First and Last Occurrences:

Code:

```

public class FirstAndLastOccurrences {
    public static void main(String[] args) {
        int[] arr = {1, 2, 2, 2, 3, 4, 5};
        int target = 2;

        int firstOccurrence = findFirstOccurrence(arr, target);
        int lastOccurrence = findLastOccurrence(arr, target);

        System.out.println("First occurrence of " + target + ": " +
            firstOccurrence);
    }
}

```

```
        System.out.println("Last occurrence of " + target + ": " +  
lastOccurrence);  
    }
```

```
public static int findFirstOccurrence(int[] arr, int target) {  
    int left = 0, right = arr.length - 1;  
    int first = -1;  
  
    while (left <= right) {  
        int mid = left + (right - left) / 2;  
        if (arr[mid] == target) {  
            first = mid;  
            right = mid - 1;  
        } else if (arr[mid] < target) {  
            left = mid + 1;  
        } else {  
            right = mid - 1;  
        }  
    }  
  
    return first;  
}
```

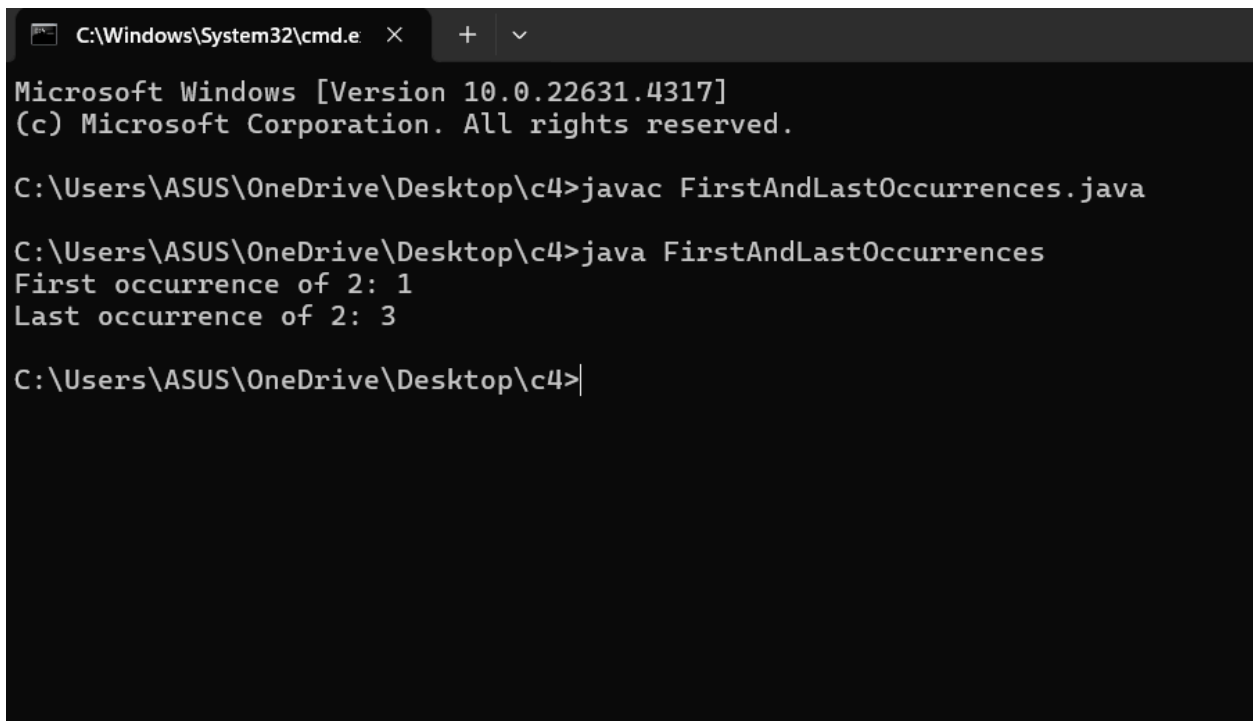
```
public static int findLastOccurrence(int[] arr, int target) {  
    int left = 0, right = arr.length - 1;  
    int last = -1;  
  
    while (left <= right) {  
        int mid = left + (right - left) / 2;  
        if (arr[mid] == target) {  
            last = mid;  
            left = mid + 1;  
        } else if (arr[mid] < target) {  
            left = mid + 1;  
        } else {  
            right = mid - 1;  
        }  
    }  
  
    return last;  
}
```

```

        right = mid - 1;
    }
}

return last;
}
}

```



The screenshot shows a Windows command prompt window with the following text:

```

C:\Windows\System32\cmd.e  X  +  v
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\OneDrive\Desktop\c4>javac FirstAndLastOccurrences.java

C:\Users\ASUS\OneDrive\Desktop\c4>java FirstAndLastOccurrences
First occurrence of 2: 1
Last occurrence of 2: 3

C:\Users\ASUS\OneDrive\Desktop\c4>

```

Time complexity:  $O(\log n)$

4) First transition point:

Code:

```

public class FirstTransitionPoint {
    public static void main(String[] args) {
        int[] arr = {0, 0, 0, 1, 1, 1};
        System.out.println("First transition point: " + findFirstTransition(arr));
    }
}

```

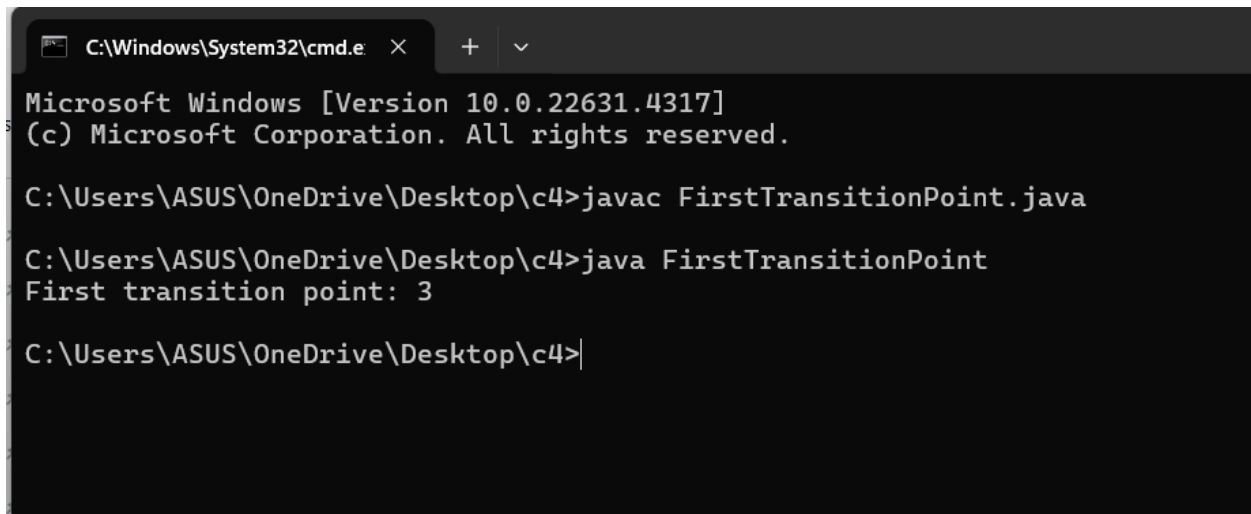
```

public static int findFirstTransition(int[] arr) {
    int left = 0, right = arr.length - 1;
    int result = -1;

    while (left <= right) {
        int mid = left + (right - left) / 2;
        if (arr[mid] == 1) {
            result = mid;
            right = mid - 1;
        } else {
            left = mid + 1;
        }
    }

    return result;
}
}

```



The screenshot shows a Windows Command Prompt window with the following text:

```

C:\Windows\System32\cmd.e  X  +  v
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\OneDrive\Desktop\c4>javac FirstTransitionPoint.java

C:\Users\ASUS\OneDrive\Desktop\c4>java FirstTransitionPoint
First transition point: 3

C:\Users\ASUS\OneDrive\Desktop\c4>

```

Time complexity:  $O(\log n)$

5) first repeating element

Code:

```

import java.util.HashSet;

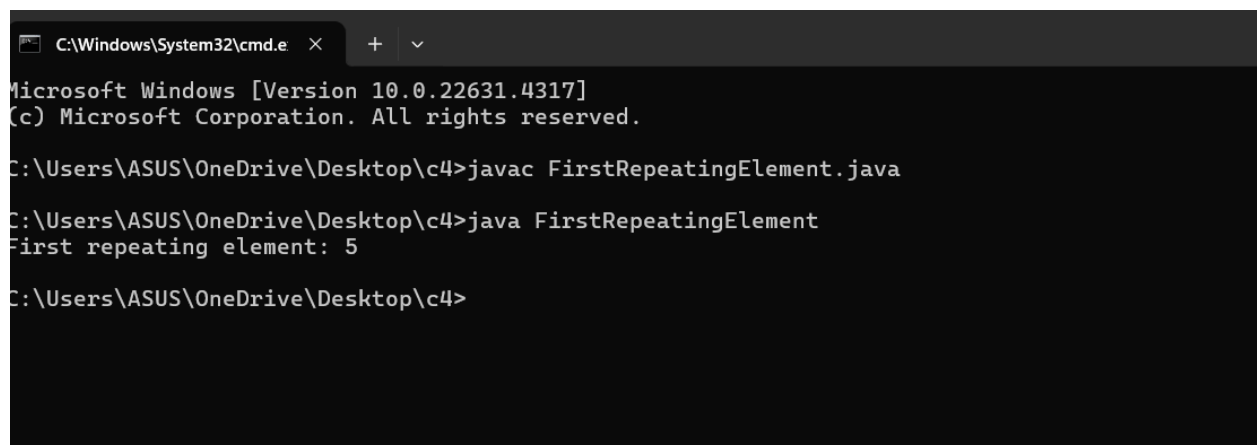
public class FirstRepeatingElement {
    public static void main(String[] args) {
        int[] arr = {4, 5, 6, 3, 5, 4, 7};
        System.out.println("First repeating element: " + firstRepeating(arr));
    }

    public static int firstRepeating(int[] arr) {
        HashSet<Integer> set = new HashSet<>();

        for (int num : arr) {
            if (set.contains(num)) {
                return num;
            } else {
                set.add(num);
            }
        }

        return -1;
    }
}

```



```

C:\Windows\System32\cmd.e  X  +  v
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\OneDrive\Desktop\c4>javac FirstRepeatingElement.java

C:\Users\ASUS\OneDrive\Desktop\c4>java FirstRepeatingElement
First repeating element: 5

C:\Users\ASUS\OneDrive\Desktop\c4>

```

Time complexity:  $O(n)$

6)Remove duplicate sorted array:

Code:

```
public class RemoveDuplicates {
    public static void main(String[] args) {
        int[] arr = {1, 1, 2, 2, 3, 4, 4, 5};
        int length = removeDuplicates(arr);

        System.out.print("Array after removing duplicates: ");
        for (int i = 0; i < length; i++) {
            System.out.print(arr[i] + " ");
        }
    }

    public static int removeDuplicates(int[] arr) {
        if (arr.length == 0) return 0;

        int j = 1;

        for (int i = 1; i < arr.length; i++) {
            if (arr[i] != arr[i - 1]) {
                arr[j] = arr[i];
                j++;
            }
        }

        return j;
    }
}
```



```
C:\Windows\System32\cmd.e  X  +  v

Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\OneDrive\Desktop\c4>javac RemoveDuplicates.java

C:\Users\ASUS\OneDrive\Desktop\c4>java RemoveDuplicates
Array after removing duplicates: 1 2 3 4 5
C:\Users\ASUS\OneDrive\Desktop\c4>
```

Time complexity:  $O(n)$

7)Maximum Index:

Code:

```
public class MaximumIndex {
    public static void main(String[] args) {
        int[] arr = {34, 8, 10, 3, 2, 80, 30, 33, 1};
        System.out.println("Maximum index difference: " + maxIndexDiff(arr));
    }

    public static int maxIndexDiff(int[] arr) {
        int n = arr.length;
        int[] leftMin = new int[n];
        int[] rightMax = new int[n];

        leftMin[0] = arr[0];
        for (int i = 1; i < n; i++) {
            leftMin[i] = Math.min(arr[i], leftMin[i - 1]);
        }
    }
}
```

```

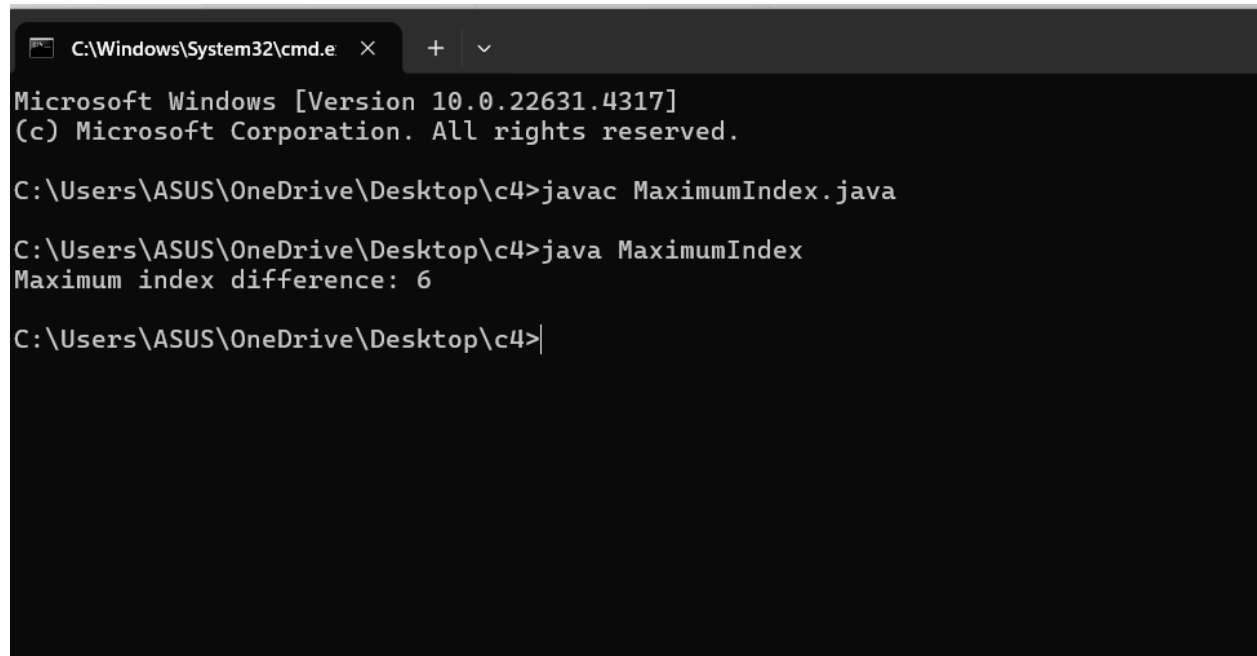
    }

    rightMax[n - 1] = arr[n - 1];
    for (int j = n - 2; j >= 0; j--) {
        rightMax[j] = Math.max(arr[j], rightMax[j + 1]);
    }

    int i = 0, j = 0, maxDiff = -1;
    while (i < n && j < n) {
        if (leftMin[i] <= rightMax[j]) {
            maxDiff = Math.max(maxDiff, j - i);
            j++;
        } else {
            i++;
        }
    }

    return maxDiff;
}
}

```



The screenshot shows a Windows Command Prompt window with the following text:

```

C:\Windows\System32\cmd.e  X  +  v
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\OneDrive\Desktop\c4>javac MaximumIndex.java

C:\Users\ASUS\OneDrive\Desktop\c4>java MaximumIndex
Maximum index difference: 6

C:\Users\ASUS\OneDrive\Desktop\c4>|

```

Time complexity:  $O(n)$

8)Wave array:

Code:

```
import java.util.Arrays;

public class WaveArray {
    public static void main(String[] args) {
        int[] arr = {1, 2, 3, 4, 5, 6};
        waveArray(arr);

        System.out.print("Wave array: ");
        for (int num : arr) {
            System.out.print(num + " ");
        }
    }

    public static void waveArray(int[] arr) {
        Arrays.sort(arr);

        for (int i = 0; i < arr.length - 1; i += 2) {

            if (i + 1 < arr.length) {
                int temp = arr[i];
                arr[i] = arr[i + 1];
                arr[i + 1] = temp;
            }
        }
    }
}
```

```
C:\Windows\System32\cmd.e  X  +  v
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\OneDrive\Desktop\c4>javac WaveArray.java

C:\Users\ASUS\OneDrive\Desktop\c4>java WaveArray
Wave array: 2 1 4 3 6 5
C:\Users\ASUS\OneDrive\Desktop\c4>
```

Time complexity:  $O(n \log n)$