

1) 0 - 1 Knapsack Problem

Code:

```
import java.util.Arrays;
```

```
class first{
    public static int getAns(int w, int idx, int[] wt, int[] val, int[][] memo) {
        if (idx < 0) return 0;
        if (memo[idx][w] != -1) return memo[idx][w];

        int a = 0;
        if (w - wt[idx] >= 0) {
            a = val[idx] + getAns(w - wt[idx], idx - 1, wt, val, memo);
        }
        a = Math.max(a, getAns(w, idx - 1, wt, val, memo));

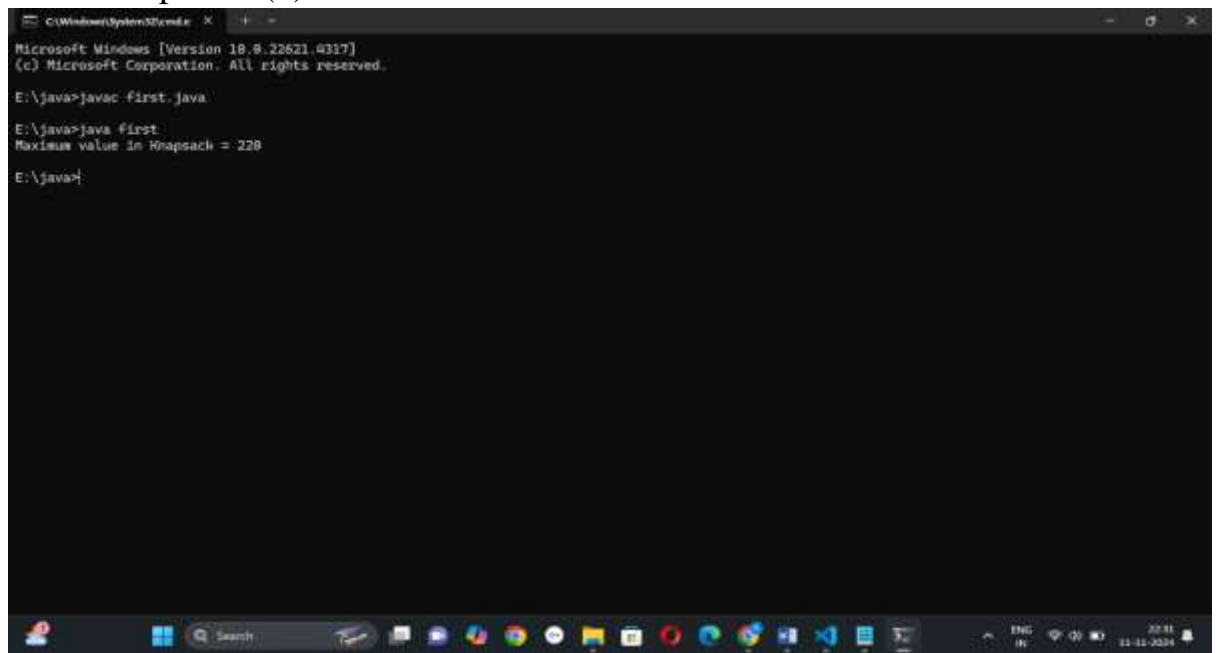
        return memo[idx][w] = a;
    }

    public static int knapSack(int W, int[] wt, int[] val) {
        int n = wt.length;
        int[][] memo = new int[n][W + 1];
        for (int[] row : memo) {
            Arrays.fill(row, -1);
        }
        return getAns(W, n - 1, wt, val, memo);
    }

    public static void main(String[] args) {
        int W = 50;
        int[] wt = {10, 20, 30};
        int[] val = {60, 100, 120};
        System.out.println("Maximum value in Knapsack = " +
            knapSack(W, wt, val));
    }
}
```

Output:

TimeComplex: $O(n)$



A screenshot of a Windows command prompt window. The title bar shows 'C:\Windows\System32\cmd.exe'. The window content displays the following text: 'Microsoft Windows [Version 10.0.22621.4317] (c) Microsoft Corporation. All rights reserved. E:\java>javac first.java E:\java>java first Maximum value in Knapsack = 220 E:\java>'. The taskbar at the bottom shows various application icons and the system clock indicating 11:11 on 11-11-2024.

2) Floor in sorted array

Code

```
class floor {

static int findFloor(int[] arr, int k) {
    int n = arr.length;
    if (arr[0] > k) return -1;

    int low = 0, high = n - 1;
    int ans = -1;

    while (low <= high) {
        int mid = low + (high - low) / 2;

        if (arr[mid] == k) {
            return mid;
        } else if (arr[mid] < k) {
            ans = mid;
            low = mid + 1;
        } else {
            high = mid - 1;
        }
    }
}
```

```

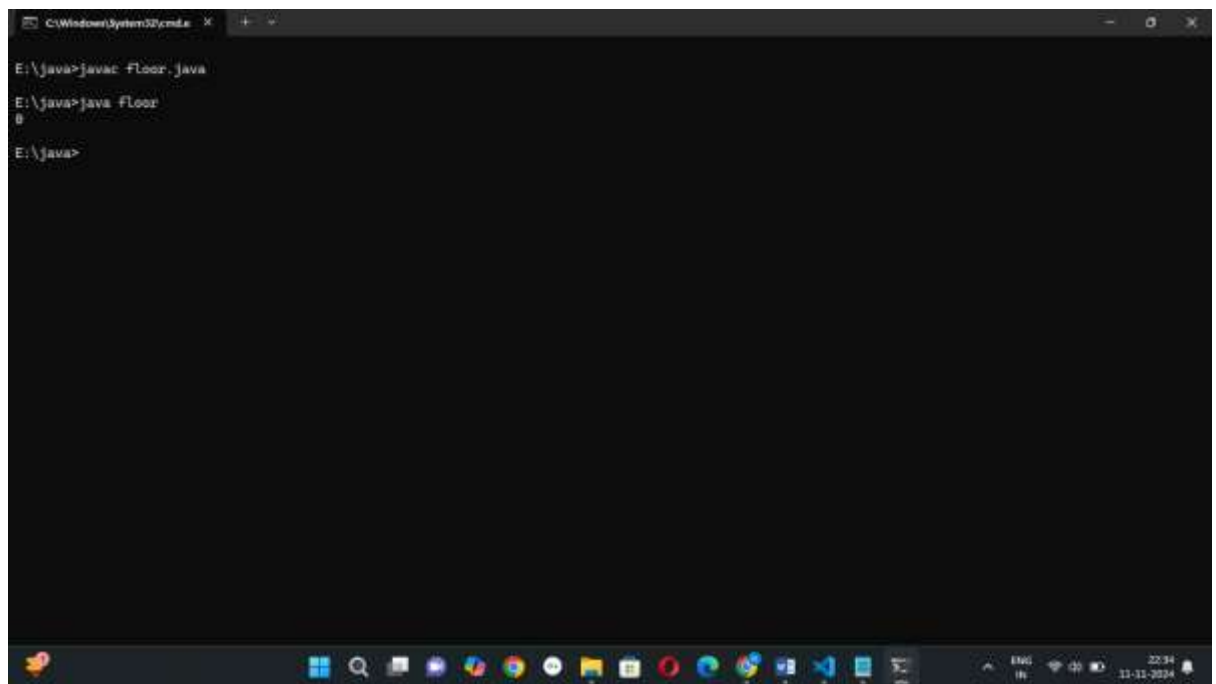
    }
    return ans;
}

public static void main(String[] args) {
    int[] arr = {1, 2, 8};
    int k = 1;
    System.out.println(findFloor(arr, k)); // Output: 0
}
}

```

Output:

Time complex:



```

C:\Windows\System32\cmd.exe
E:\java> javac floor.java
E:\java> java floor
0
E:\java>

```

3) Palindrome linked list

Code :

```
import java.util.Stack;
```

```

class Node {
    int data;
    Node next;
}

```

```

Node(int data) {
    this.data = data;
    this.next = null;
}
}

```

```

public class palindrome{
    public static boolean isPalindrome(Node head) {
        Stack<Integer> stack = new Stack<>();
        Node temp = head;

        while (temp != null) {
            stack.push(temp.data);
            temp = temp.next;
        }

        temp = head;
        while (temp != null) {
            if (temp.data != stack.pop()) {
                return false;
            }
            temp = temp.next;
        }
        return true;
    }
}

```

```

public static void main(String[] args) {
    int[] arr = {1, 2, 2, 1};

    Node head = new Node(arr[0]);
    Node current = head;
    for (int i = 1; i < arr.length; i++) {
        current.next = new Node(arr[i]);
        current = current.next;
    }
}

```

```

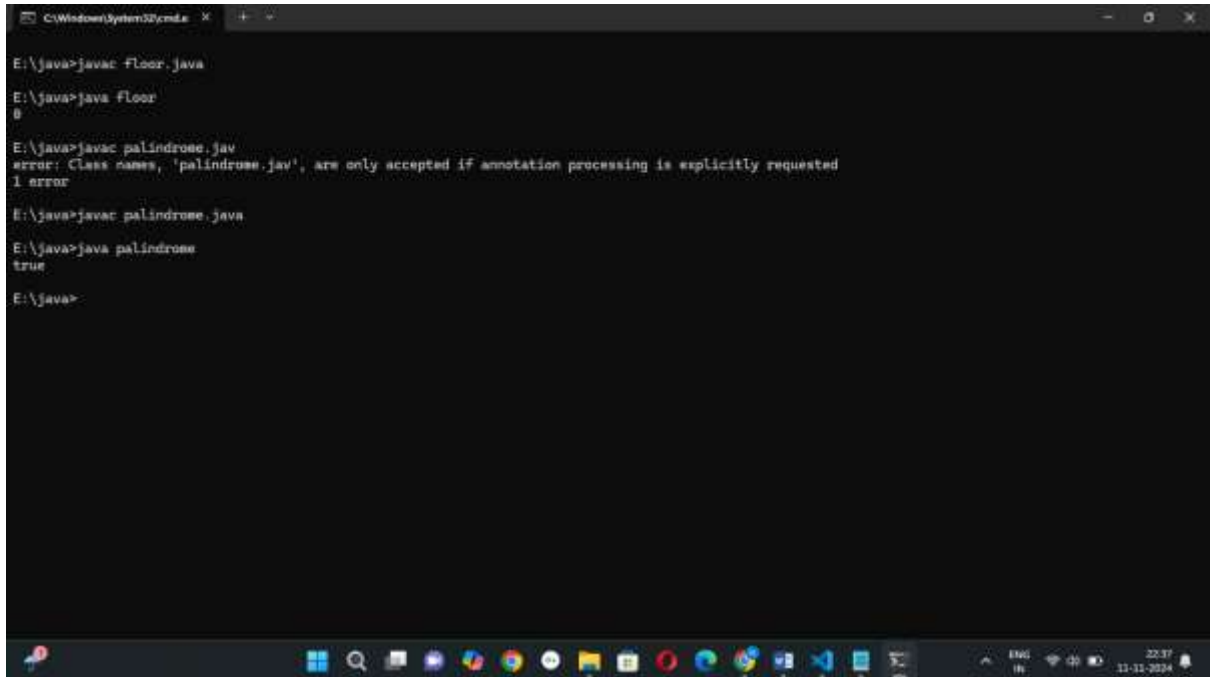
    }

    System.out.println(isPalindrome(head));
}
}

```

OutPut:

Tc: $O(n)$



```

C:\Windows\System32\cmd.exe
E:\java>javac floor.java
E:\java>java floor
0
E:\java>javac palindrome.java
error: Class names, 'palindrome.java', are only accepted if annotation processing is explicitly requested
1 error
E:\java>javac palindrome.java
E:\java>java palindrome
true
E:\java>

```

4) Balanced tree check

Code:

```

class TreeNode {
    int data;
    TreeNode left;
    TreeNode right;

    TreeNode(int data) {
        this.data = data;
    }

    TreeNode(int data, TreeNode left, TreeNode right) {

```

```
    this.data = data;
    this.left = left;
    this.right = right;
}
}
```

```
class checktree{

    public static boolean check(TreeNode root) {
        return treecheck(root) != -1;
    }

    public static int treecheck(TreeNode root) {
        if (root == null) {
            return 0;
        }

        int lf = treecheck(root.left);
        if (lf == -1) {
            return -1;
        }

        int rt = treecheck(root.right);
        if (rt == -1) {
            return -1;
        }

        if (Math.abs(lf - rt) > 1) {
            return -1;
        }

        return 1 + Math.max(lf, rt);
    }
}
```

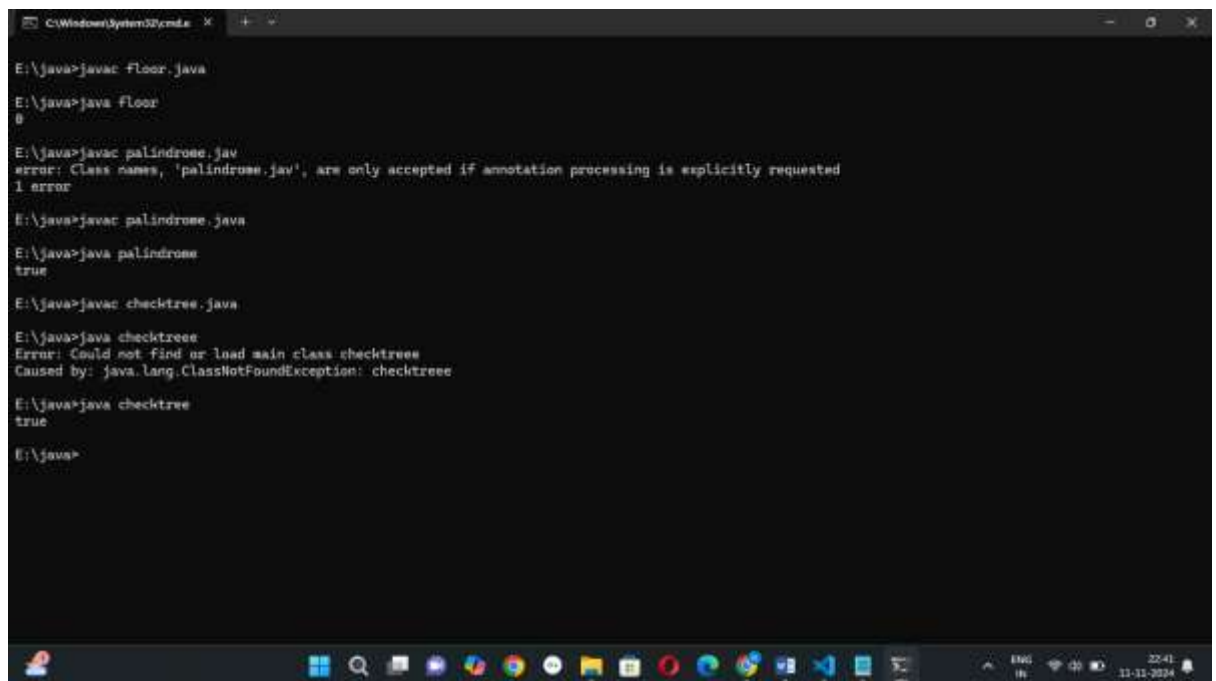
```

    public static void main(String[] args) {
        TreeNode root = new TreeNode(1);
        root.left = new TreeNode(2);
        root.right = new TreeNode(3);
        System.out.println(check(root));
    }
}

```

Output:

TC: $O(n)$



```

C:\Windows\system32\cmd.exe
E:\java>javac floor.java
E:\java>java floor
0

E:\java>javac palindrome.java
error: Class names, 'palindrome.java', are only accepted if annotation processing is explicitly requested
1 error
E:\java>javac palindrome.java
E:\java>java palindrome
true

E:\java>javac checktree.java
E:\java>java checktree
Error: Could not find or load main class checktree
Caused by: java.lang.ClassNotFoundException: checktree

E:\java>java checktree
true

E:\java>

```

5) Triplet sum in array

Code:

```

import java.util.Arrays;

class threesum{
    static boolean find3Numbers(int[] arr, int sum) {
        int n = arr.length;

```

```

Arrays.sort(arr);

for (int i = 0; i < n - 2; i++) {
    int l = i + 1;
    int r = n - 1;

    while (l < r) {
        int curr_sum = arr[i] + arr[l] + arr[r];
        if (curr_sum == sum) {
            System.out.println("Triplet is " + arr[i] + ", " + arr[l] + ", " +
arr[r]);

            return true;
        } else if (curr_sum < sum) {
            l++;
        } else {
            r--;
        }
    }
}

return false;
}

public static void main(String[] args) {
    int[] arr = { 1, 4, 45, 6, 10, 8 };
    int sum = 22;
    find3Numbers(arr, sum);
}
}

```


Output:

TC: $O(n)$

```
C:\Windows\System32\cmd.exe

E:\java>javac floor.java
E:\java>java floor
0

E:\java>javac palindrome.jav
error: Class names, 'palindrome.jav', are only accepted if annotation processing is explicitly requested
1 error
E:\java>javac palindrome.java
E:\java>java palindrome
true

E:\java>javac checktree.java
E:\java>java checktree
Error: Could not find or load main class checktree
Caused by: java.lang.ClassNotFoundException: checktree

E:\java>java checktree
true

E:\java>javac threesum.java
E:\java>java threesum
Triplet is 0, 8, 10

E:\java>
```