DSA Problems

11/11/2024

1. 0-1 Knapsack Problem:

Code:

```java
import java.util.Scanner;

public class Knapsack {
    public static int knapSack(int W, int wt[], int val[], int n) {
        int[][] dp = new int[n + 1][W + 1];

        for (int i = 0; i <= n; i++) {
            for (int w = 0; w <= W; w++) {
                if (i == 0 || w == 0)
                    dp[i][w] = 0;
                else if (wt[i - 1] <= w)
                    dp[i][w] = Math.max(val[i - 1] + dp[i - 1][w - wt[i - 1]], dp[i - 1][w]);
                else
                    dp[i][w] = dp[i - 1][w];
            }
        }
        return dp[n][W];
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number of items:");
        int n = sc.nextInt();
        System.out.println("Enter knapsack capacity:");
        int W = sc.nextInt();
        int[] val = new int[n];
        int[] wt = new int[n];

        System.out.println("Enter values:");
        for (int i = 0; i < n; i++) val[i] = sc.nextInt();
        System.out.println("Enter weights:");
        for (int i = 0; i < n; i++) wt[i] = sc.nextInt();

        System.out.println("Maximum value in Knapsack = " + knapSack(W, wt, val, n));
    }
}

```

Output:

```
Enter number of items:
3
Enter knapsack capacity:
50
Enter values:
60 100 120
Enter weights:
10 20 30
Maximum value in Knapsack = 220
```

2. Floor in Sorted Array:

Code:

```java
import java.util.Scanner;

public class Floor {
    public static int findFloor(int arr[], int n, int x) {
        int low = 0, high = n - 1, res = -1;
        while (low <= high) {
            int mid = (low + high) / 2;
            if (arr[mid] == x)
                return mid;
            else if (arr[mid] < x) {
                res = mid;
                low = mid + 1;
            } else {
                high = mid - 1;
            }
        }
        return res;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter number of elements:");
        int n = scanner.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter sorted array:");
        for (int i = 0; i < n; i++) arr[i] = scanner.nextInt();
        System.out.println("Enter value of x:");
        int x = scanner.nextInt();
        int result = findFloor(arr, n, x);
        System.out.println("Floor of " + x + " is at index " + result);
    }
}
```

Output:

```
Enter number of elements:
5
Enter sorted array:
1 2 6 7 10
Enter value of x:
7
Floor of 7 is at index 3
```

3. Check Equal Arrays:

Code:

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

public class Equalarray {
    public static boolean areEqual(int[] arr1, int[] arr2) {
        Arrays.sort(arr1);
        Arrays.sort(arr2);
        return Arrays.equals(arr1, arr2);
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter size of arrays:");
        int n = sc.nextInt();
        int[] arr1 = new int[n];
        int[] arr2 = new int[n];

        System.out.println("Enter elements of first array:");
        for (int i = 0; i < n; i++) arr1[i] = sc.nextInt();

        System.out.println("Enter elements of second array:");
        for (int i = 0; i < n; i++) arr2[i] = sc.nextInt();

        if (areEqual(arr1, arr2))
            System.out.println("Arrays are equal");
        else
            System.out.println("Arrays are not equal");

            sc.close();
    }
}

```

Output:

```
Enter size of arrays:
5
Enter elements of first array:
1 2 3 4 5
Enter elements of second array:
 2 4 6 8 10
Arrays are not equal
```

4. Palindrome Linked List:

Code:

```java
import java.util.Scanner;

class ListNode {
    int val;
    ListNode next;
    ListNode(int val) { this.val = val; }
}

public class PalindromList {
    public static boolean isPalindrome(ListNode head) {
        StringBuilder sb = new StringBuilder();
        ListNode curr = head;
        while (curr != null) {
            sb.append(curr.val);
            curr = curr.next;
        }
        return sb.toString().equals(sb.reverse().toString());
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number of elements in linked list:");
        int n = sc.nextInt();
        ListNode head = new ListNode(sc.nextInt());
        ListNode curr = head;

        for (int i = 1; i < n; i++) {
            curr.next = new ListNode(sc.nextInt());
            curr = curr.next;
        }

        if (isPalindrome(head))
            System.out.println("Linked list is a palindrome");
        else
            System.out.println("Linked list is not a palindrome");
    }
}
```

Output:

```
Enter number of elements in linked list:
3
1 2 4
Linked list is not a palindrome
PS F:\java> ^C
PS F:\java>
PS F:\java>  f:; cd 'f:\java'; & 'C:\Progra
\Roaming\Code\User\workspaceStorage\41ec956
Enter number of elements in linked list:
3
1 2 1
Linked list is a palindrome
```

5. **Triplet Sum in Array**:

Code:

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

public class Triplet {
    public static boolean findTriplet(int arr[], int n, int sum) {
        Arrays.sort(arr);
        for (int i = 0; i < n - 2; i++) {
            int l = i + 1, r = n - 1;
            while (l < r) {
                if (arr[i] + arr[l] + arr[r] == sum)
                    return true;
                else if (arr[i] + arr[l] + arr[r] < sum)
                    l++;
                else
                    r--;
            }
        }
        return false;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number of elements:");
        int n = sc.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter array elements:");
        for (int i = 0; i < n; i++) arr[i] = sc.nextInt();
        System.out.println("Enter target sum:");
        int sum = sc.nextInt();

        if (findTriplet(arr, n, sum))
            System.out.println("Triplet exists");
        else
            System.out.println("No triplet found");
    }
}

```

Output:

```
Enter number of elements:
6
Enter array elements:
1 2 38 6 10 8
Enter target sum:
22
No triplet found
```