1-D Dynamic Programming Solutions (Java, Recursion)

```
import java.util.*; public class NCR { static final int MOD = 1000000007; static int solve(int n, int r, int[][] dp) { if (r == 0 || r == n) return 1; if (dp[n][r] != -1) return dp[n][r]; return dp[n][r] = (solve(n - 1, r - 1, dp) + solve(n - 1, r, dp)) % MOD; } public static void main(String[] args) { Scanner sc = new Scanner(System.in); int n = sc.nextInt(), r = sc.nextInt(); int[][] dp = new int[n + 1][r + 1]; for (int[] row : dp) Arrays.fill(row, -1); System.out.println(solve(n, r, dp)); } }
```

```
import java.util.*; public class Tribonacci { static int solve(int n, int[] dp) { if (n == 0) return 0; if (n == 1 || n == 2) return 1; if (dp[n] != -1) return dp[n]; return dp[n] = solve(n - 1, dp) + solve(n - 2, dp) + solve(n - 3, dp); } public static void main(String[] args) { Scanner sc = new Scanner(System.in); int n = sc.nextInt(); int[] dp = new int[n + 1]; Arrays.fill(dp, -1); System.out.println(solve(n, dp)); } }
```

```
import java.util.*; public class MinCostClimb { static int solve(int[] cost, int i, int[] dp) { if (i >= cost.length) return 0; if (dp[i] != -1) return dp[i]; return dp[i] = cost[i] + Math.min(solve(cost, i + 1, dp), solve(cost, i + 2, dp)); } public static void main(String[] args) { Scanner sc = new Scanner(System.in); int n = sc.nextInt(); int[] cost = new int[n]; for (int i = 0; i < n; i++) cost[i] = sc.nextInt(); int[] dp = new int[n]; Arrays.fill(dp, -1); System.out.println(Math.min(solve(cost, 0, dp), solve(cost, 1, dp))); } }
```

```
import java.util.*; public class Boredom { static long solve(int i, long[] freq, long[] dp) { if (i <= 0) return 0; if (dp[i] != -1) return dp[i]; return dp[i] = Math.max(solve(i - 1, freq, dp), solve(i - 2, freq, dp) + i * freq[i]); } public static void main(String[] args) { Scanner sc = new Scanner(System.in); int n = sc.nextInt(); long[] freq = new long[100001]; for (int i = 0; i < n; i++) freq[sc.nextInt()]++; long[] dp = new long[100001]; Arrays.fill(dp, -1); System.out.println(solve(100000, freq, dp)); } }
```