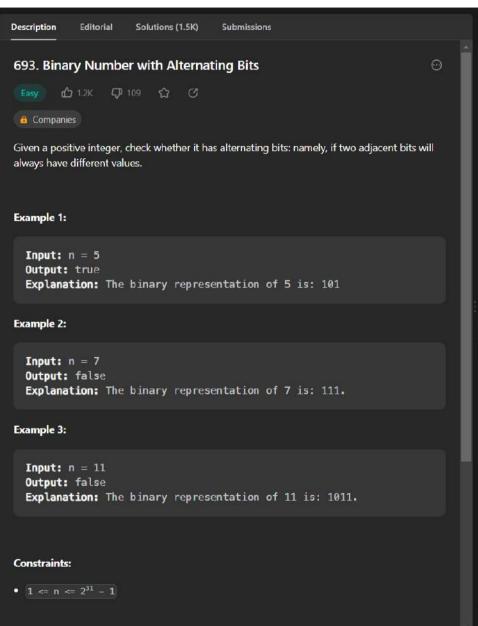


```
i Java V - Auto
     public class Solution {
         int[][] dp;
         int finish:
         public int solve(int[] locations, int currCity, int remainingFuel) {
             if (remainingFuel < 0) {
                 return 0;
             if (dp[currCity][remainingFuel] != -1) {
                  return dp[currCity][remainingFuel];
             int ans = currcity == finish ? 1 : 0;
             for (int nextCity = 0; nextCity < n; nextCity++) {
                  if (nextCity != currCity) {
                      ans = (ans + solve(locations, nextCity,
                              remainingFuel - Math.abs(locations[currCity] - locations[nextCity]))) %
     1000000007;
         Result
Testcase
Accepted Runtime: 0 ms

    Case 1

              • Case 2
                          · Case 3
Input
  [2,3,6,8,4]
  1
Console Y
                                                                                                      Submit
                                                                                            Run
```



```
i Java 🗸 🕒 - Auto
         public boolean hasAlternatingBits(int n) {
             String s = Integer.toBinaryString(n);
             char prev = s.charAt(0);
             for(int i=1; i<s.length(); i++) {
                 int x = Character.compare(prev, s.charAt(i));
                 System.out.println(x);
                 if(x == 0) {
                 prev = s.charAt(i);
 15 }
Testcase Result
Accepted Runtime: 0 ms
  · Case 1
              • Case 2
                          • Case 3
Input
  5
Stdout
Console Y
                                                                                           Run
                                                                                                    Submit
```