11 June. Date : __/__/__

1. Longest Balanced Substring of 0s and 1s.

Approach: Prefix sum + HashMap.

convert 0 to -1.

Maintain a running sum (prefix sum)

use a HashMap.

If prefix sum repeats, means the substring b/w the two occurrences has a sum of 0.

Java Code:

```
Public class LongestSubstring (String s) {
  Map< Integer, Integer > map = new HashMap<>();
    int sum = 0; maxLength = 0;
     int n = s.length();
  for (int i = 0; i<n, i++) {
    int val = (s.charAt(i) == '0') ? -1 : 1;
    sum += val;
    if ( sum == 0) {
      maxLength = i+1; }
 if (map.containsKey(sum)) {
 maxLength = Math.max(maxLength, i - map.get(sum));
   } else {
    map.put(sum, i);
   }
  }
 return maxLength;
 }
}
```

[illegible]:

2. Maximize Minimum Distance Between Gas Stations.

Approach: Binary search

→ Possible minimum distance lies b/w 1 and P[N-1]-P[0]

→ use Binary Search to find the largest minimum distance d such that at least k stations can be placed with distance ≥ d.

Java code:

```
        public static boolean isPossible (int[] P, int N, int K,
                int dist) {.
        int count = 1;
          int lastPos = P[0];
      for (int i = 1; i < N; i++) {.
          if( P[i] - lastPos >= dist) {.
            Count ++;
            lastPos = P[i];
            if(count == k)  return true;
              }
          } return false;
        }.
  Public static int maxMinDist( int[] P, int N, int K) {.
            int low = 1;
          Int high = P[N-1] - P[0];
            int ans = 0;
        while ( low <= high) {.
            int mid = low + (high - low) / 2;
            if (isPossible (P, N, K, mid)) {
                ans = mid;
        low = mid +1;
        } else {
            high = mid-1;
         }
        }

        return ans;
     }
   }
```

Dry Run : low = 1 ; high = 16.

→ mid = 8.

count = 2.

increas low = 9.

→ mid = 12.

count = 2.

increas low = 13

→ mid = 14

2nd station at 17.

count = 2, increase low = 15

→ mid = 15

count = 2.

increase low = 16.

→ mid = 16.

count = 16.

→ increase low = 17.

exit loop low = 17, high = 16.

final ans = 16, but sample output is 9.

→ At mid = 9, 12-1 = 11 ≥ 9. , 17-12 = 5 < 9

count = 2

max feasible min distance before failing is 9.