1. Find index of no of 1 and no of 0Find max subarray with max sum - Kadan's algorithm

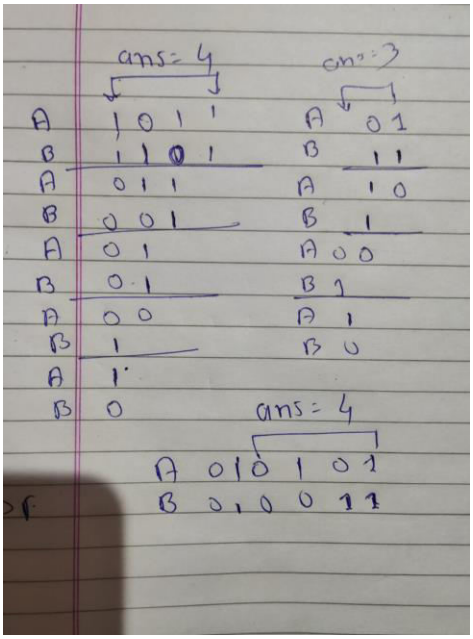
```
int max_so_far = INT_MIN, max_ending_here = 0;

for (int i = 0; i < size; i++)
{
    max_ending_here = max_ending_here + a[i];
    if (max_so_far < max_ending_here)
        max_so_far = max_ending_here;

    if (max_ending_here < 0)
        max_ending_here = 0;
}
return max_so_far;
```

1. Binary XOR (CODEcheck long challange)
   https://www.codechef.com/DEC19B/problems/BINXOR
   Find the all possible hamming distance between two strings
   Min_hd = ones(A) + ones(B)
   Max_hd = ones(A) - ones(B)

   For x = Min_hd to Max_hd ; x += 2
   Ans += n! / (x! * (n-x)!)

   Eg : A = 001111 A = 11110
   B = 000011 B = 11100
   Possible hd = 6, 4, 2, 0 hd = 3, 1

2. Binary Addition (codechef long challange)
   https://www.codechef.com/DEC19B/problems/BINADD/
   Find the max diff between index of two ones' and nearest two ones' or two zeros'



3. Trie
    a. https://leetcode.com/problems/implement-trie-prefix-tree/
    b. https://leetcode.com/problems/add-and-search-word-data-structure-design/ (trie with regular expression .)
4. Union find
    a. https://leetcode.com/problems/most-stones-removed-with-same-row-or-column/submissions/
    b. code

```
for(int i=0;i<1201;i++)
        parent[i] = i;
int parent[1201];

  int find(int u)
  {
     if(parent[u] == u)
        return u;
     int x = find(parent[u]);
     parent[u] = x;
     return x;
  }

  void dounion(int u, int v)
  {
     int pu = find(u);
     int pv = find(v);

     if(pu == pv)
```

```
            return;

        parent[pu] = pv;
    }
```

5. https://leetcode.com/problems/trapping-rain-water/

6. Intervals
    a. https://leetcode.com/problems/merge-intervals/submissions/

7. Division without using / * mod

   https://leetcode.com/problems/divide-two-integers/