

Tries for bits of numbers

MAXXOR AND MINXXOR

```
typedef struct data
{
    data *bit[2];
    int cnt = 0;
} btrie;

btrie *head;

void insert(int x)
{
    btrie *node = head;
    for (int i = 30; i >= 0; i--)
    {
        int cbit = (x >> i) & 1;
        if (node->bit[cbit] == NULL)
        {
            node->bit[cbit] = new btrie();
        }
        node = node->bit[cbit];
        node->cnt++;
    }
}

void remove(int x)
{
    btrie *node = head;
    for (int i = 30; i >= 0; i--)
    {
        int cbit = (x >> i) & 1;
        node = node->bit[cbit];
        node->cnt--;
    }
}

int maxxor(int x)
{
    btrie *node = head;
    int ans = 0;
    for (int i = 30; i >= 0; i--)
    {
        int cbit = (x >> i) & 1;
        //MATCH OPP BITS FOR MAXXOR
        if (node->bit[cbit ^ 1] != NULL && node->bit[cbit ^ 1]->cnt > 0)
        {
            ans += (1LL << i);
            node = node->bit[cbit ^ 1];
        }
    }
}
```

```

        else
            node = node->bit[cbit];
    }
    return ans;
}

int minxor(int x)
{
    btrie *node = head;
    int ans = 0;
    for (int i = 30; i >= 0; i--)
    {
        int cbit = (x >> i) & 1;
        //MATCH SAME BITS FOR MINXOR
        if (node->bit[cbit] != NULL && node->bit[cbit]->cnt > 0)
            node = node->bit[cbit];
        else
        {
            ans += (1LL << i);
            node = node->bit[cbit ^ 1];
        }
    }
    return ans;
}

int main()
{
    kira;
    int t, x;
    char op;
    cin >> t;
    head = new btrie();
    //Maxxor then insert(0) not for minxor
    insert(0);
    while (t--)
    {
        cin >> op >> x;
        if (op == '+')
            insert(x);
        else if (op == '-')
            remove(x);
        else
        {
            int ans = maxxor(x);
            p1(ans);
        }
    }
    return 0;
}

```

MAXXOR SUBARRAY:

Insert prefix xors in the trie and query for maxxor for each subarray ending with a[i]

