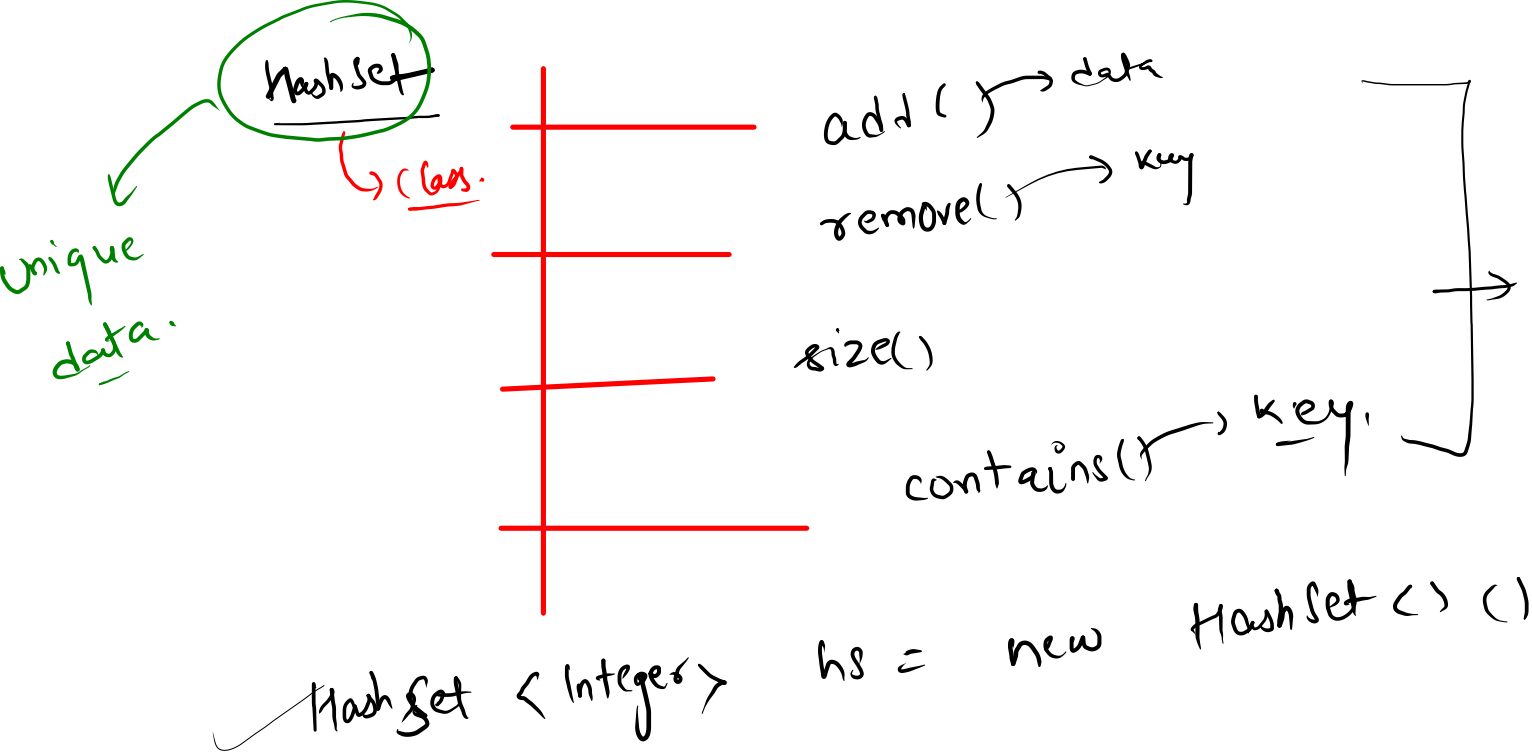

HashSet
class.
Unique data.
add( ) → data
remove( ) → key
size( )
contains( ) → key.
HashSet <Integer> hs = new HashSet<>();

409. Longest Palindrome

Easy 4855 309 Add to List Share

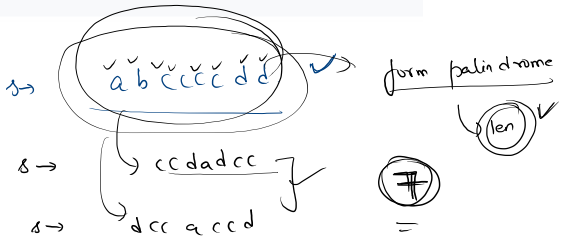
Given a string `s` which consists of lowercase or uppercase letters, return *the length of the* ***longest palindrome*** that can be built with those letters.

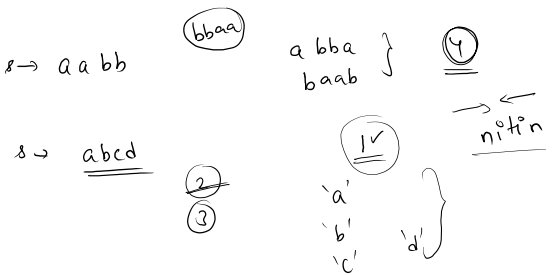
Letters are **case sensitive**, for example, "Aa" is not considered a palindrome here.

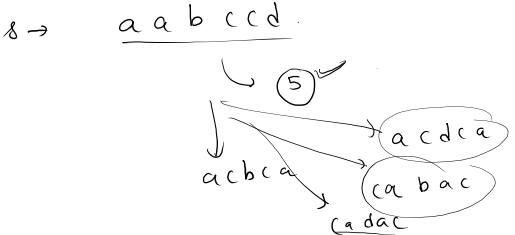
ab

Example 1:

```
Input: s = "abccccdd"
Output: 7
Explanation: One longest palindrome that can be built is "dccaccd", whose
length is 7.
```





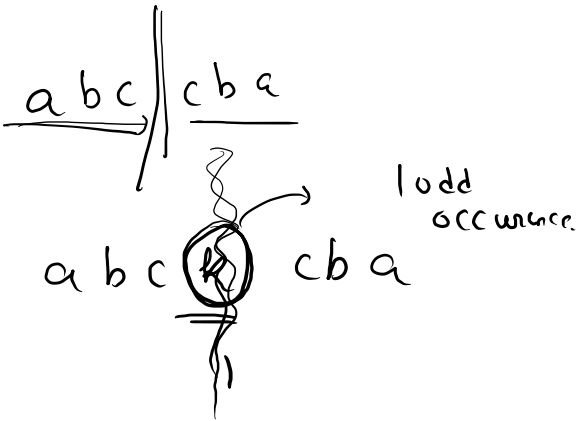


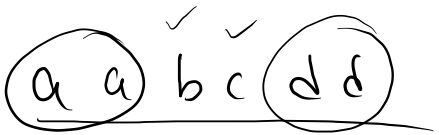
s → a b cccc dd

ans = 7.

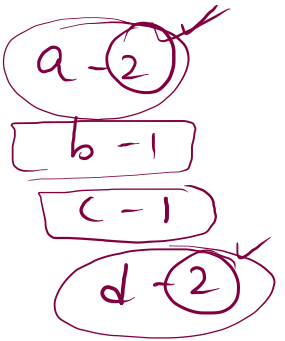
2 palindrome → even

odd.





→ ans → 5



d a b c a d

da ad

ans = ~~4~~ 5.

a a b b c c →

a	2
b	2
c	2

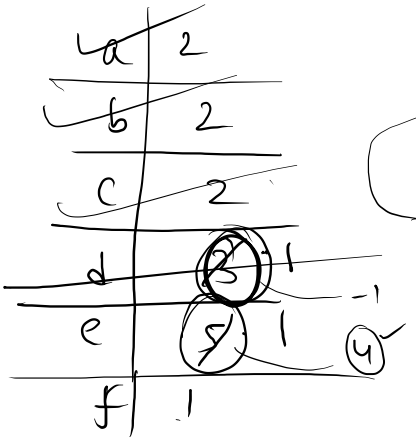
ans = 6.

---

a a b b c c d d d e e e e e f

ans = 13.

ee d c b a d a b c d ee



n → odd → 13

nearest smaller even

→ 12

```java
class Solution {
    public int longestPalindrome(String s) {
        HashMap<Character, Integer> hm = new HashMap<>();

        for(int i = 0; i < s.length(); i++){
            char ch = s.charAt(i);

            hm.put(ch, hm.getOrDefault(ch, 0) + 1);
        }
        int ans = 0;
        boolean oddPresent = false;
        for(Character key: hm.keySet()){
            int val = hm.get(key);

            if(val % 2 == 0){
                ans += val;
            }
            else{
                ans += val - 1;
                oddPresent = true;
            }

        }
        if(oddPresent){
            ans++;
        }
        return ans;
    }
}
```

aa bb cc ddd eeeee f

a	2
b	2
c	2
d	3
e	5
f	1

ans = 0 2 4 6 8 12

odd = false. → true. 13

key = a b c d e f

val = 2 2 2 3 5 1

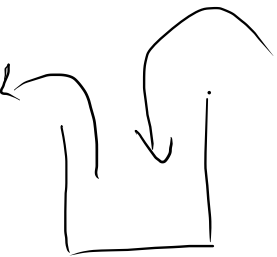
Queue ⇝ pipe like DS. → open from 2 ends

⇝ FIFO (First in first out)

Real life.

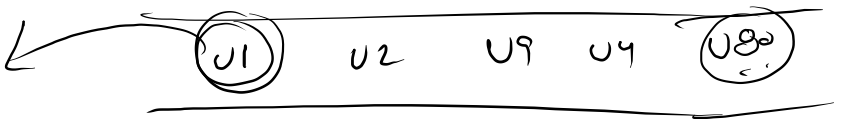
↳ Booking

↳ irctc



72 seats

1000



Queue → Interface

implement → Class

- → ArrayDeque
- → Linkedlist
- → Priority Queue

Queue AD. → +

↳ ArrayList

~~Array Deque~~

TATKAL

A B C D

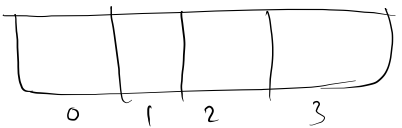
Senior Citizen

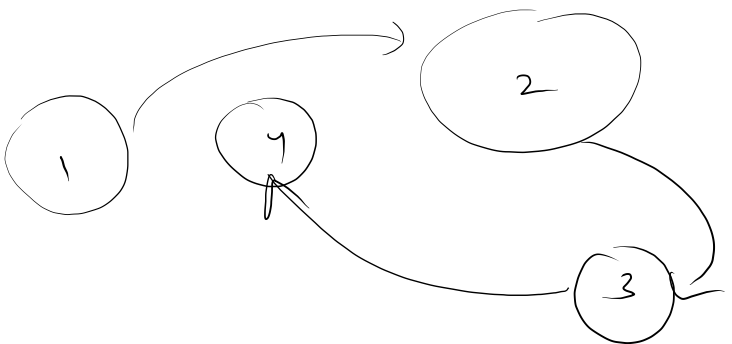
72 lower

List → AL

Queue → AD LL PQ

↳ easy & fast.





Queue.

```java
import java.util.Queue;
import java.util.LinkedList;
public class Main
{
    public static void main(String[] args) {
        Queue<Integer> qu = new LinkedList<>();

        qu.add(10);
        qu.add(20);
        qu.add(30);
        qu.add(40);

        System.out.println(qu.peek());

        qu.remove();

        System.out.println(qu);
        System.out.println(qu.size());

    }
}
```

Queue Syntax Learning

1. **Declare an Empty** $queue\ s$.
2. Take Single Integer T as input.
3. For next T Lines format $(case, x(optional))$

- case 1. $Print$ the $size$ of the $queue$ in a separate line.
- case 2. $Remove$ an element from the queue. If the queue is empty then print -1 in a separate line.
- case 3. Add Integer x to the $queue\ s$.
- case 4. $Print$ an element at the $front$ of the $queue$. If queue is empty print -1 in a seperate line.

Sample Input 0

```
5
1
2
3 9
4
1
```

Sample Output 0

```
0
-1
9
1
```

```
public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();

        Queue<Integer> qu = new LinkedList<>();

        for(int i = 0; i < n; i++){
            int caseNumber = scn.nextInt();

            if(caseNumber == 1){
                //size
                System.out.println(qu.size());
            }
            else if(caseNumber == 2){
                if(qu.size() == 0){
                    System.out.println(-1);
                }
                else{
                    qu.remove();
                }
            }
            else if(caseNumber == 3){
                int x = scn.nextInt();  //optional
                qu.add(x);
            }
            else{
               if(qu.size() == 0){
                    System.out.println(-1);
                }
                else{
                    System.out.println(qu.peek());
                }
            }
        }

    }
}
```

Decimal Number Sys.

0 ---- 9 ⇒ 10 digits

Binary Number Sys.

0/1

0

1

$(10)_2 \rightarrow 2$

$(11)_2 \rightarrow 3$

Print Binary.

Input:

N = 5

Output:

1 10 11 100 101

5

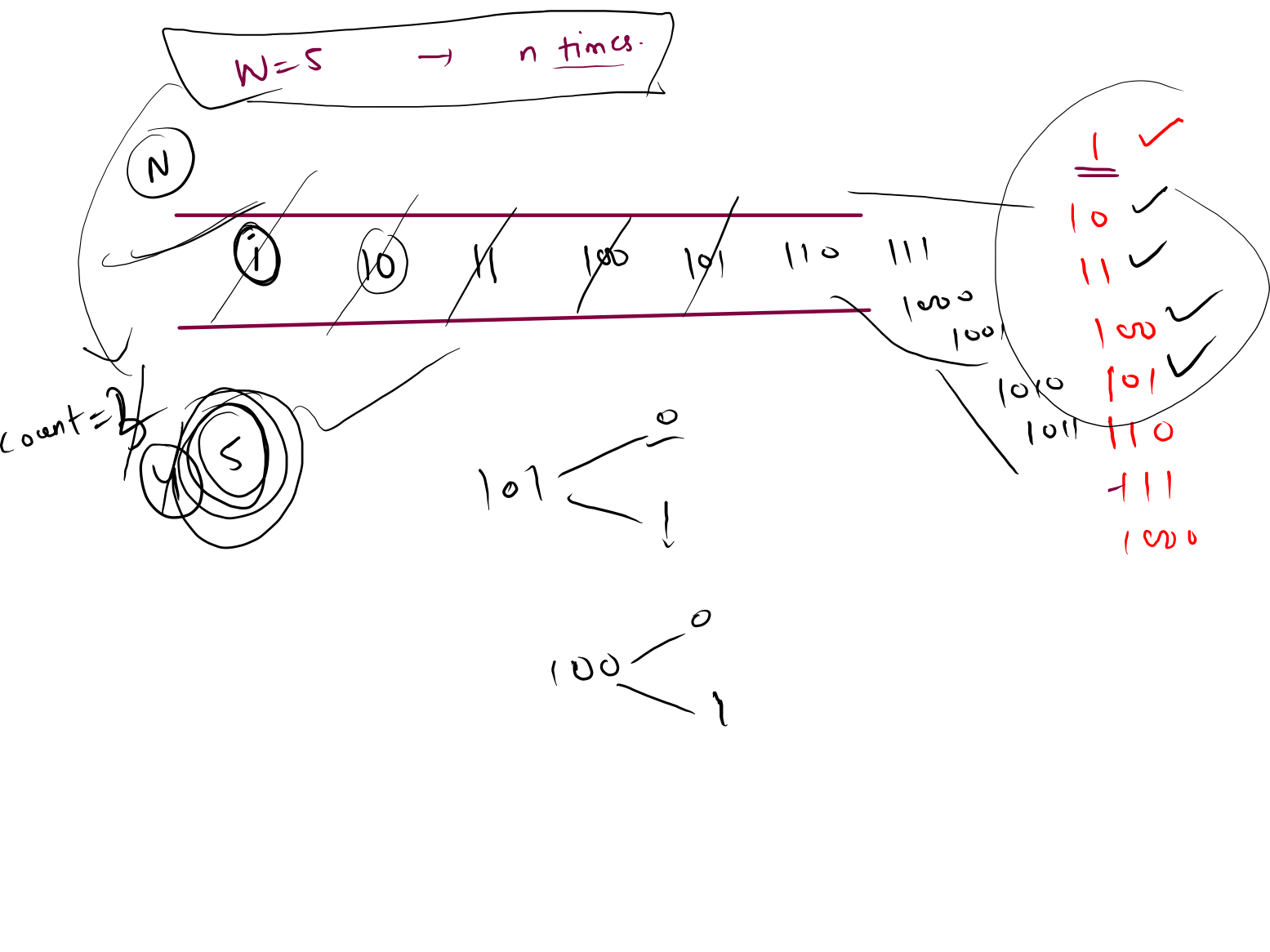
N=4

1 10 11 100

N=6

1 10 11 100 101 110

0
1
10
11
100
101
110
111
1000


W=5 → n times.
N
1
10
11
100
101
110
111
1000
1001
1010
1011
count=2
S
101
0
1
100
0
1

N=4

1 10 11 100

```
39
40  class solve{
41
42      //Function to generate binary numbers from 1 to N using a queue.
43      static ArrayList<String> generate(int N)
44      {
45          ArrayList<String> ans = new ArrayList<>();
46          Queue<String> qu = new LinkedList<>();
47          qu.add("1");
48
49          for(int i = 0; i < N; i++){
50              String rem = qu.remove();
51              ans.add(rem);
52
53              qu.add(rem + "0");
54              qu.add(rem + "1");
55          }
56          return ans;
57      }
58
59  }
60
```

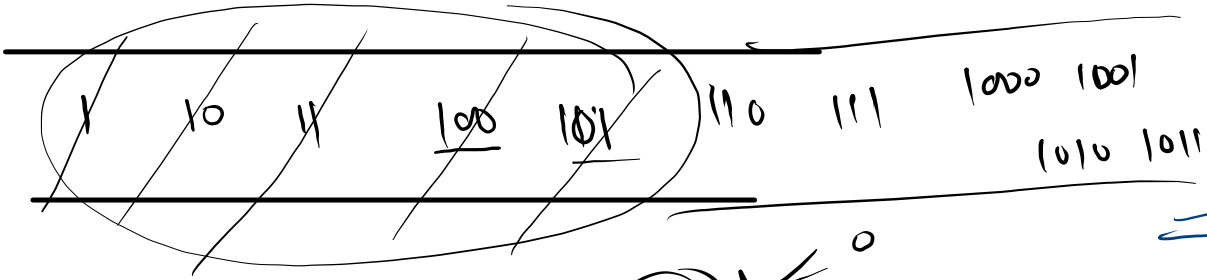
Syso(rem)

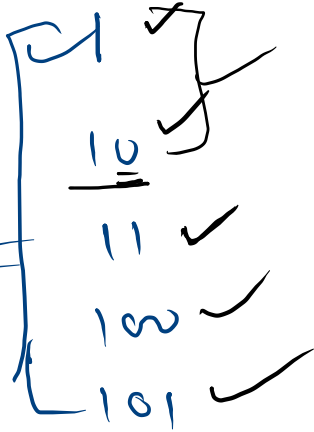
N=4.

100 101

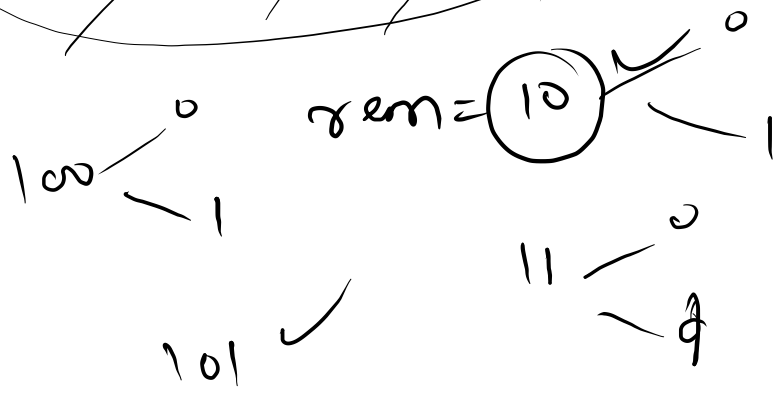
rem = 11

N=5


1
10
11
100
101
110
111
1000
1001
1010
1011


10
11
100
101
110
111
1000

rem = 10


0
100
1
101
0
1
11
0
9

First Negative Integer 2

```
5 2
-8 2 3 -6 10
```

Sample Output 0

```
-8 0 -6 -6
```