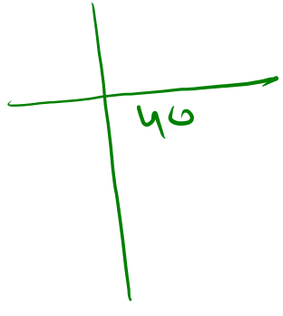
n = 4

4x4

saddle

- row min
- col max

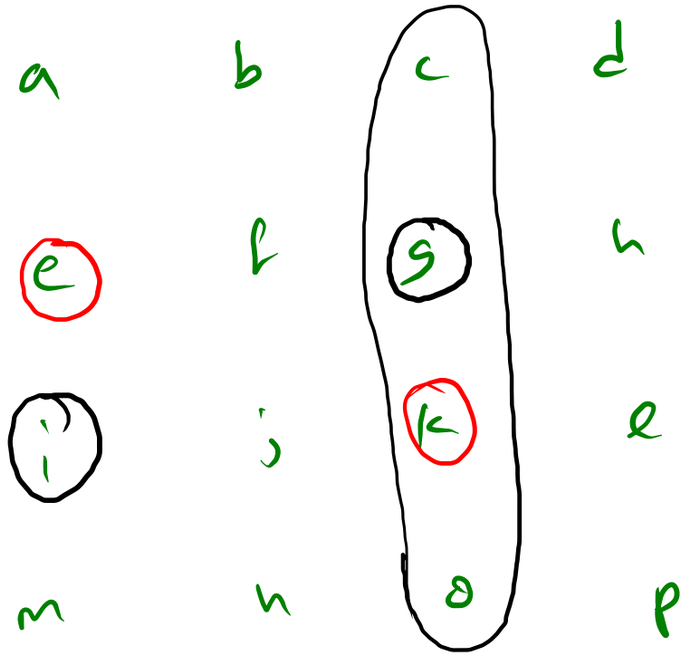


0

10	20
20	10

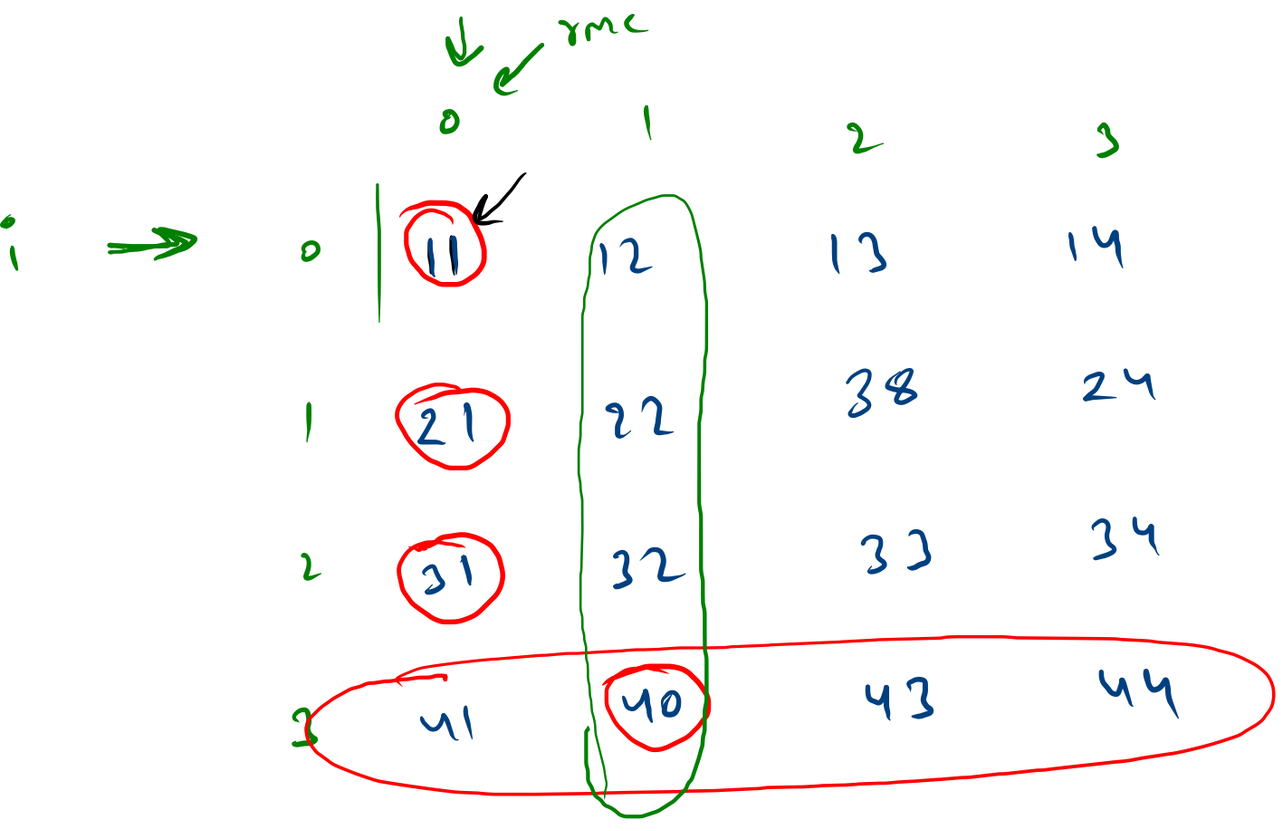
Invalid Input

```
4 11 12 13 14 21
22 23 24 31 32
33 34 41 42 43
44
```



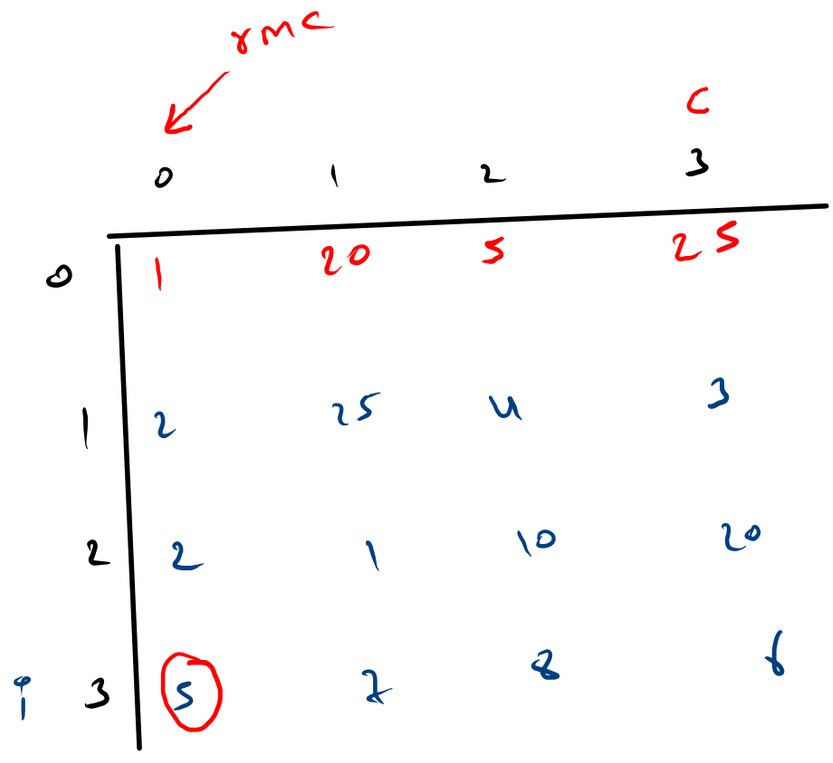
$i < e < g \rightarrow i < g$ ✓

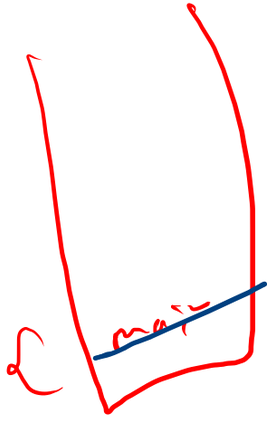
$g < k < i \rightarrow g < i$ ✗




```
i ← 0 to n-1
    rmc = 0
    // row min col
    int flag = 1;
    j ← 0 do n-1
        if( ar[0][rmc] > ar[i][j]
            flag=0;

if( flag==1)
    print ar[i][rmc]
```

```
for(int i=0;i<n;i++){
    int rmc = 0;
    for(int c=0;c<n;c++){
        if(arr[i][c] < arr[i][rmc]){
            rmc = c;
        }
    }

    int flag = 1;
    for(int r=0;r<n;r++){
        if(arr[r][rmc] > arr[i][rmc]){
            flag = 0;
            break;
        }
    }

    if(flag == 1){
        System.out.println(arr[i][rmc]);
        return;
    }
}

System.out.println("Invalid input");
```





Strings

String s1 = "abcABC123#?";

0 1 2

s1.length();

s1.charAt(2);

include ↓ exclude ↓

substring(i , j)

2 1

s1 = " a b c d " (0 1 2 3 4 5)

0, 0 → ""

0, 1 → "a"

0, 2 → "ab"

0, 3 → "abc"

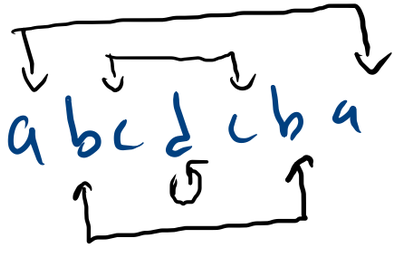
0, 4 → "abcd"

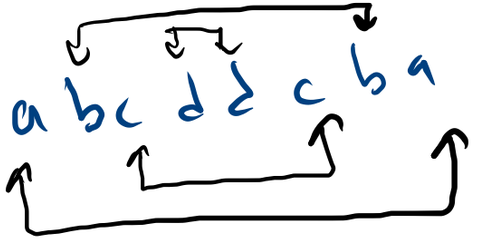
0, 5 → ⚠

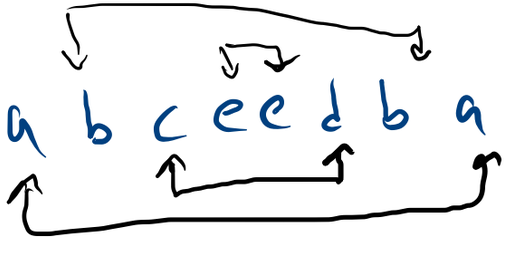
string s = s1 . substring(i , j)

1, 3 → "bc"

strings is palindrome







s1.length()
substring(i, j)
charAt(i);

0 1 2 3 4
a b c c

- a
- b
- c
- cc
- c

0,1 a ✓
0,2 a b
0,3 a b c
0,4 a b cc

1,2 b ✓
1,3 b c
1,4 b cc

2,3 c ✓
2,4 c c ✓

3,4 c ✓

0 → 1 ⋮ 4

1 → 2 ⋮ 4

2 → 3 ⋮ 4

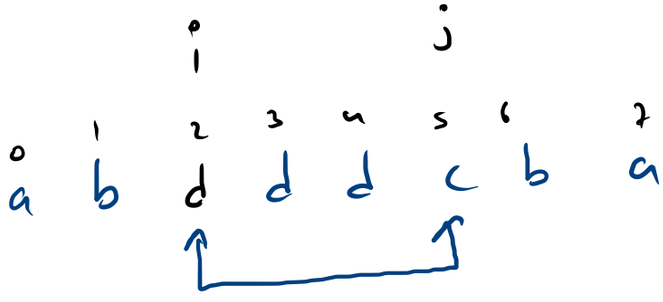
3 → 4 ⋮ 4

* [all substring's
[is palindrome

i ← 0 do n-1

j ← i+1 to n

i < j



j i

a b c b a

i == j

n = 8

```
while ( i<j ) {
    char a = charAt(i)
    char b = charAt(j)

    if ( a == b) {
        i++; j--;
    } else {
        return false;
    }
}
return true;
```

```java
public static boolean isPalindrome(String ss){
    int n = ss.length();

    int i=0;
    int j = n-1;

    while(i<j){
        char a = ss.charAt(i);
        char b = ss.charAt(j);
        if(a == b){
            i++;
            j--;
        }else{
            return false;
        }
    }
    return true;
}
```

i j

ss = a b c c b a

n = 6

i = 0 1 2 3

j = 5 4 3 2

i < j

j < 2

a = 'c'

b = 'c'

str 2 **aaabbccdee** a a aa

a b c d e a

compression 1

compression 2 a3 b2 c2 d e2 a4

aaabbccdee

i
j

string ans = "";

while(i<n) {

[while (j<n && charAt(i) == charA(j))
j++;

ans = ans + charAt(i);

i=j

}

ans= "abcde"

i

j

aaabbccdee

count = 0 + 2 3

count = 1

ans = a 3 b 2 c 2 d

String ans = "";

while (i < n) {

int count = 0;

[while (j < n && charAt(i) == charAt(j))
j++; count++;

ans = ans + charAt(i);
if (count > 1)
ans = ans + count;

} i = j

String s1 = "abc";

String s2 = "abc";

String s3 = new String("abc");

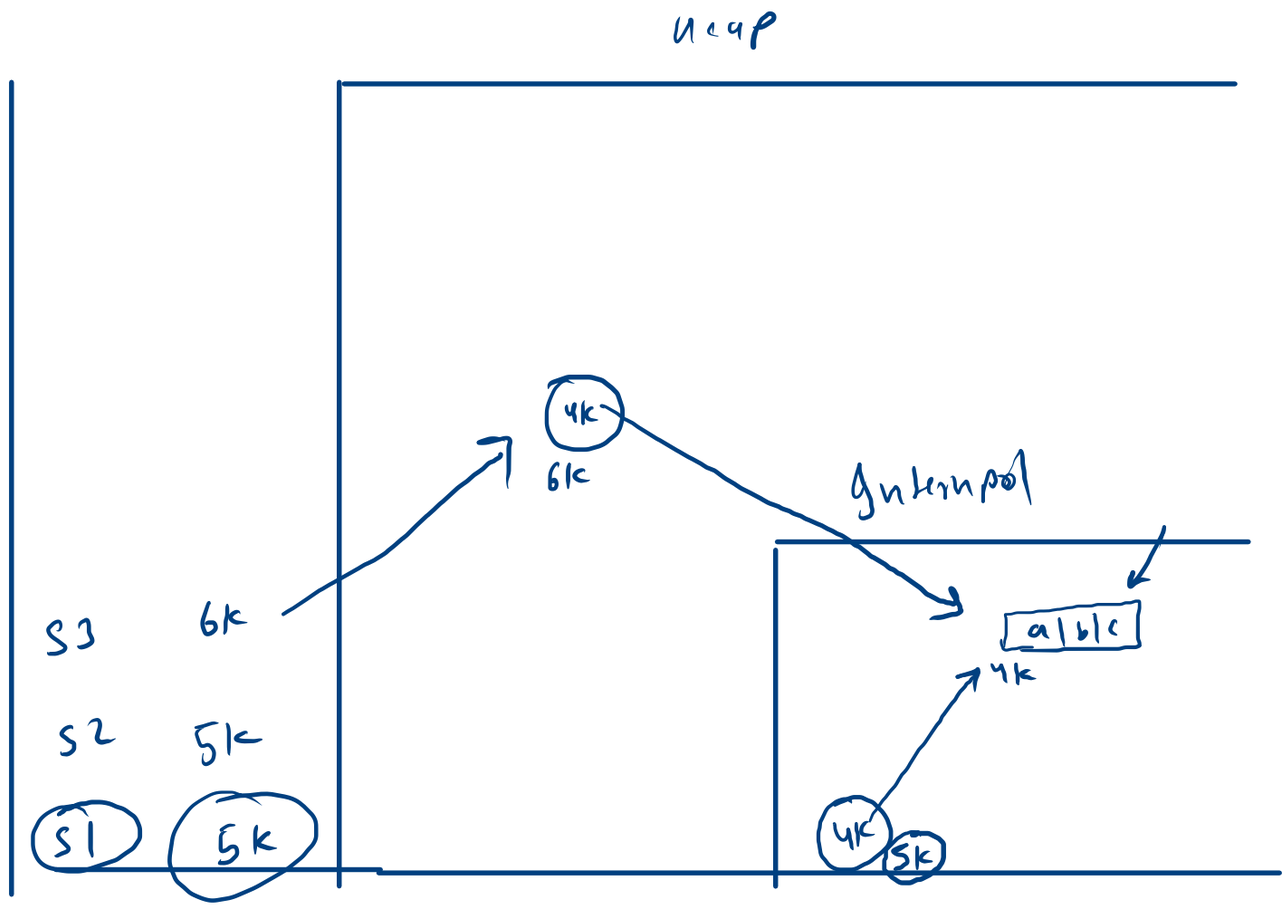
charAt(i)

s1.setCharAt(i, 'g');

if (s1 == s2) True

s1 == s3 False

s1.equals(s3) True



String s = ""

for (i = 0 ; i < 5 ; i++) {

s = s + i ;

}

s = s + "0" ;

s = "0" + 1

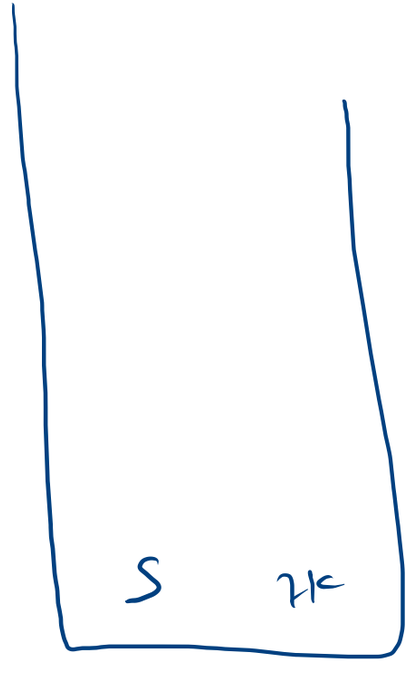
StringBuilder

set

append

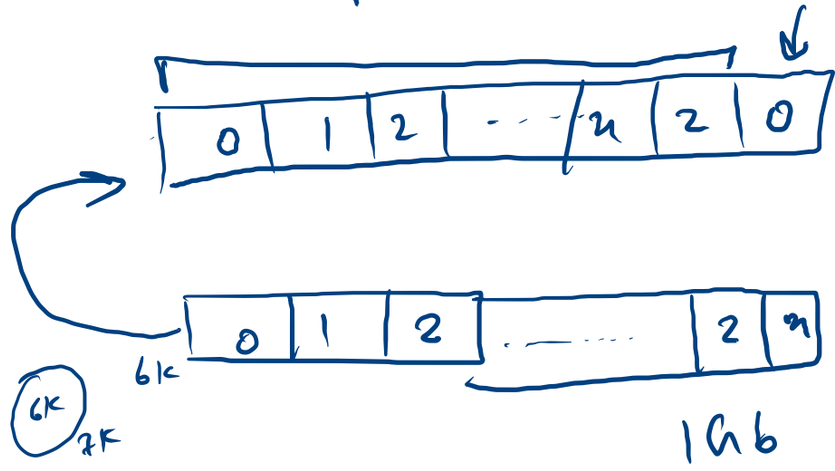
insert

delete



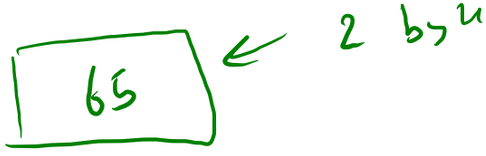
n

1 Gb



1 Gb


```
char ch = s.charAt(i)

ch = 'A';

(char) (ch + 1)
```




```
ch >= 'A' && ch <= 'Z'

'a' - 'A' = 32
```

'0' → 48

'9'

'A' → 65

'B' → 66

'Z' →

'a' → 97

'b' → 98

'z' →

string s = sb.toString()

pepCODinG