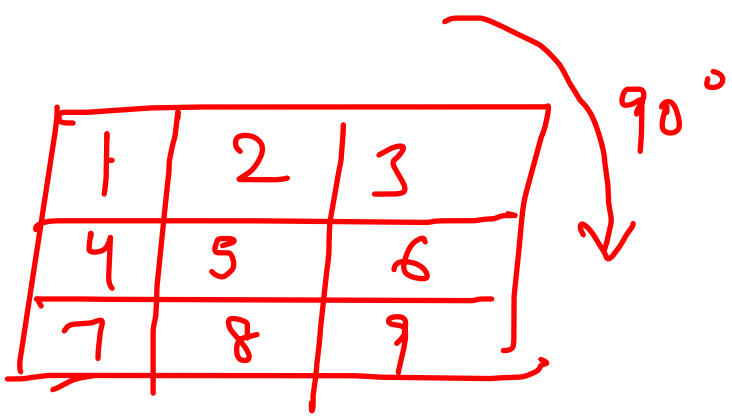
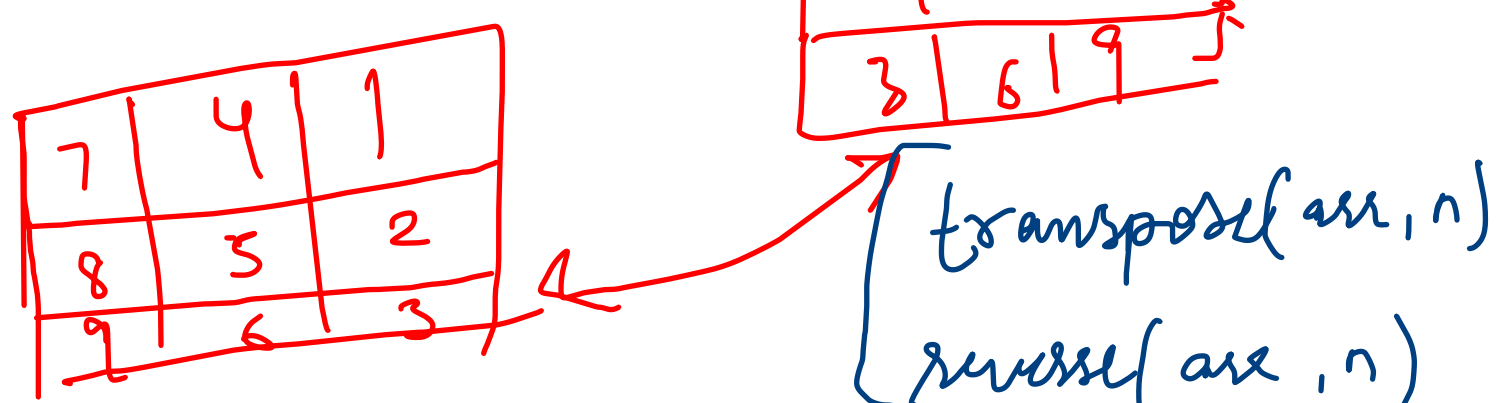
```
//transpose
for(i=0; <n; i++){
for(j=0; <n; j++){
if (i > j)
temp = arr[i][j]
arr[i][j] = arr[j][i]
arr[j][i] = temp
```

```
//reverse
for( i=0; <n; i++){
   si=0 , ei = n-1
while ( si < ei){
      swap( ——— )
      si++;
      ei--;
```



abcba → complete

cfabagh → aba

abbc → bb

→ longest ✓
→ palindrome ✓
→ substring ✓

| | SC | TC |
|---|---|---|
| → Brute force | ✓ O(1) | ✓ $O(n^3)$ |
| → DP | $O(n^2)$ | $O(n^2)$ |
| → Expand from centre ✓✓ | ✓ O(1) | ✓ $O(n^2)$ |
| → Manchar's Algo | $O(n)$ | $O(n)$ |

Brute

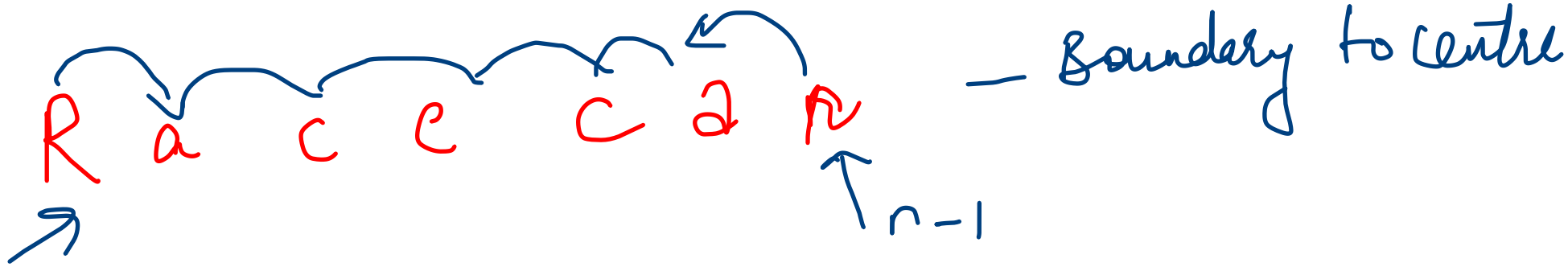
"raceca r"

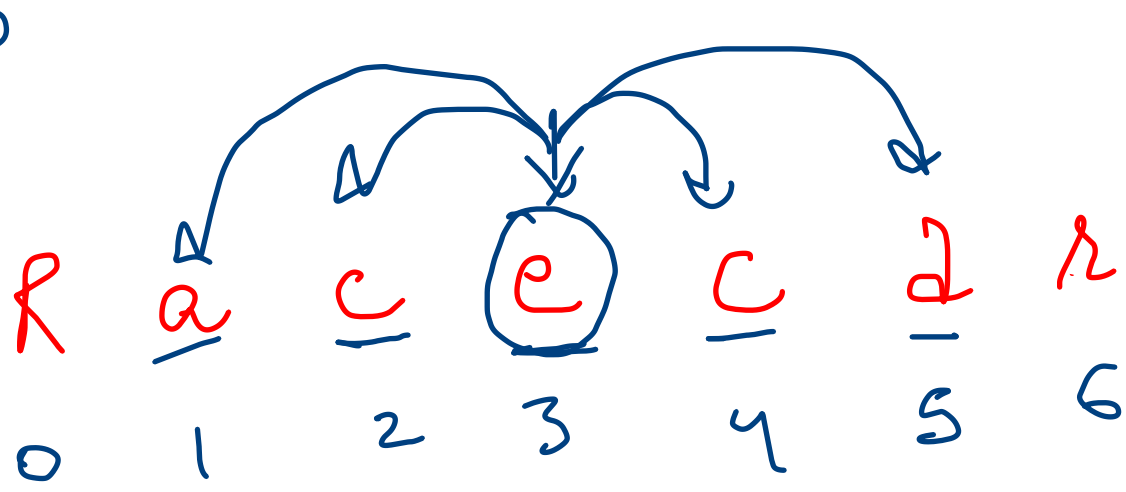
ra ✗ racec

race ✗ racecar

raceca ✗

racecar




$l = i, 3$

$r = i, 3$

a b b d

$l$ $r$

$l = i$

$r = i+1$

r a c e c a r
0 1 2 3 4 5 6
↓l,r

n = 7

add length

3 e == e T

2,4 c == c T

1,5 a == a T

0,6

max = 1

start = 0

end = 0

```
for(int i = 0; < n-1; i++)
```

i = 0

```
l = i
r = i
```

```
while ( l >= 0 && r < n)
    if ( charAt(l) == charAt(r))
        l--;
        r++;
    }
    else{
        break;
```

length = r - l + 1

Lm > max

max = Lm

start = l+1

end = r-1

```java
public static String longest(String str){

    int max  = 1;
    int n = str.length();

    int start = 0;
    int end = 0;

    // odd length

    for(int i = 0; i < n-1; i++){
        int l = i;
        int r = i;

        while(l >= 0 && r < n){
            if(str.charAt(l) == str.charAt(r)){
                l--;
                r++;
            }
            else
                break;
        }

        int length = r-l+1;
        if(length > max){
            max = length;

            start = l +1;
            end = r-1;
        }
    }

    // even length

    for(int i = 0; i < n-1; i++){
    int l = i;
    int r = i+1;

    while(l >= 0 && r < n){
        if(str.charAt(l) == str.charAt(r)){
            l--;
            r++;
        }
        else
            break;

    }
        int length = r-l+1;
        if(length > max){
            max = length;

            start = l +1;
            end = r-1;
        }
    }
        return str.substring(start, end+1);
}
```

$TC - O(n^2)$

$SC - O(1)$

Str : " Race car "
0 1 2 3 4 5 6

maxL = 1
n = 7
S = 0
e = 0

i = 0, r
l = 0
r = 0
(l >= 0 && r < n)
(0 >= 0 && 0 < 7) T
r == r T
l = -1
r = 1
length = r - (-1) + 1 = 1

l = -1, r = 7

i = 1, a
l = 1
r = 1
(1 >= 0 && 1 < 7) T
a == a T
l = 0
r = 2

i = 2, c
l = 2
r = 2
( ) T
c == c T
l = 1
r = 3
a == e X

i = 3, e
l = 3
r = 3
( ) T
c == e T

l = 2, r = 4
c == c T
l = 1, r = 5
a == a T
l = 0, r = 6
r == r T

length = 7 - (-1) + 1

= 7

7 > 1

max = 7

start = -1+1 = 0

end = 7-1 = 6

Substring (0, 7) → 0-6

Raccar

→ abbd

→ Radar

→ bdaaa

→ aabbcc