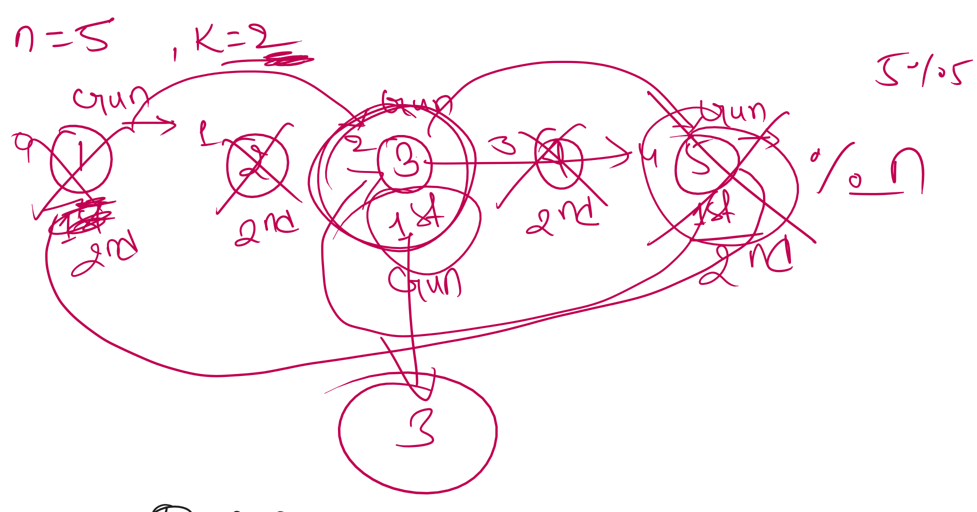
leetcode : 1823

Josephus Problem

||

Find the winner of Circular Game

n = 5 , K = 2



5 % 5 = 0

% n

```
int solve (int n, int k){
   if(n==1) return 0;
   return (solve(n-1,k) +k) % n;
}
```

1 2 3 4 5 == 1

1 ~~2~~ 3 4 5 % n

(n-1 , k) + k % n

index of value candidate

solve(5,2) => 2

(solve(4,2) + 2) % 5
0+2 % 5 = 2

(solve(3,2) + 2) % 4
2+2 % 4 = 0

(solve(2,2) + 2) % 3
0+2 % 3 = 2

(solve(1,2) + 2) % 2
0 + 2 % 2 = 0

```
int findTheWinner (int n, int k){
     return  solve(n,k) +1;
}
```

2+1

3

Time Complexity :

1. Best case ($\Omega(n)$)
2. Average case ($\Theta(n)$)
3. Worst case ($O(n)$)

① a = 10 , b = 20 ①

temp = a → ①

a = b → ①

b = temp → ①

$O(1+1+1+1+1)$

$O(5) \Rightarrow O(1)$

constant time complexity

a — ①

b — ①

temp — ①

$S(1+1+1)$

$S(1)$

n = 5

```
for(i=1; i<=n; i++){      (n+1)
    printf("%d", i);
}
```

$O(n)$

$S(n)$ || $O(n)$

```
for(i=1; i<=n; i++){      (n+1)
    printf("%d", i);
}
for( j=1; j<=n; j++){  →  (n+1)
    printf("%d", j);
}
```

n+1 + n+1

2n+2

2(n+1)

$O(2n)$

```
for(i=1; i<=n; i++){ →  n+1
   for( j=1; j<=n; j++){  n*(n+1)
     printf(   );
   }
}
```

i = 1,

j = 1,2,3,4,5

$n+1 + n^2+n$

n^2+2n+1

$O(n^2)$

```
for(i=1; i<=n; i++){ →  n+1
  for( j=1; j<=n; j++){  n*(n+1)
    for( k=1; k<=n; k++){  n*n*(n+1)
      printf("%d");
    }
  }
}
```

$n+1 + n^2+n + n^3+n^2$

$n^3 + 2n^2 + 2n + 1$

Binary Search

key - 1

l = 0   100%   h = 4

1	3	5	7	9

mid-1   2   mid+1

mid = (l+h)/2 = $\frac{0+4}{2}$ = 2

$\log(n)$

100% 50% 50% 10%

1	3	2

7	9

l   mid-1 mid   mid+1 l   h

mid = $\frac{l+h}{2}$ mid = $\frac{l+h}{2}$

Divide & conquer

n log(n)

Merge sort

Quick sort

2

```
      0 1 2 3 4
arr[] = {1,2,3,4,5}
rev[n] ;
for( i=0; i<n; i++){
   5 4 3 2 1 5   X
rev[i] = arr[n-1-i]
}
```

rev[] = {5, 4, 3, 2, 1} (indices 0, 1, 2, 3, 4)

rev[0] = arr[5-1-0] = arr[4]

rev[1] = arr[5-1-1] = arr[3]

rev[2] = arr[5-1-2] = arr[2]

rev[3] = arr[5-1-3] = arr[1]

rev[4] = arr[5-1-4] = arr[0]

```
arr[] = {1,2,3,4,5}
d = 2
for( i=0; i<n; i++){
  ~~5 4 3 2 1 0 < 5~~ X
  rotated[i] = arr[(i+d)%n]
}
```

rotated[] = {3 4 5 1 2} (indices 0, 1, 2, 3, 4)

3 4 5 1 2

(1+2) % 5

(2+2) % 5 = 4

rotated[0] = arr[(0+2)%5] = arr[2]

rotated[1] = arr[3]

rotated[2] = arr[4]

rotated[3] = arr[(3+2) % 5)

rotated[4] = arr[((4+2) % 5] = arr[1]

```
Integer f1(Integer n, Integer r)
if(n > 0)
return (n - r + f1(n/3, 13))
else
return 0
end if
```

0 > 0

n = 39   r = 13

→ f1(39, 13)

if(39 > 0)

return (39-13 + f1 (13, 13))

26 + f1 (13, 13)   (n, r)

→ f1(13, 13)

if(13 > 0)

26 + ~~13 - 13~~ + f1 (4, 13) => 26 + f1(4, 13)

f1 (4, 13)

4 > 0

26 + 4 - 13 + f1(1 , 13)

17 + f1 (1, 13)   (n, r)

1 > 0

17 + 1 - 13 + f(0, 13)

5

5 + f(0, 13)