📂 4. Painting Fence Algorithm

Problem Statement:
The painting fence algorithm determines the number of ways to paint a fence with multiple 'N' posts and 'K' colours. The algorithm ensures that at most 2 adjacent posts (no more than two adjacent posts) have the same colour. Since answer can be large return it modulo 10^9 + 7 (1000000007).

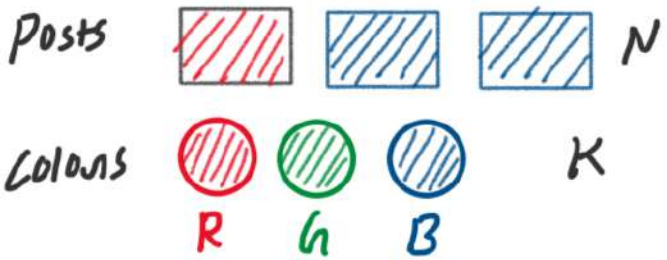
Examples:

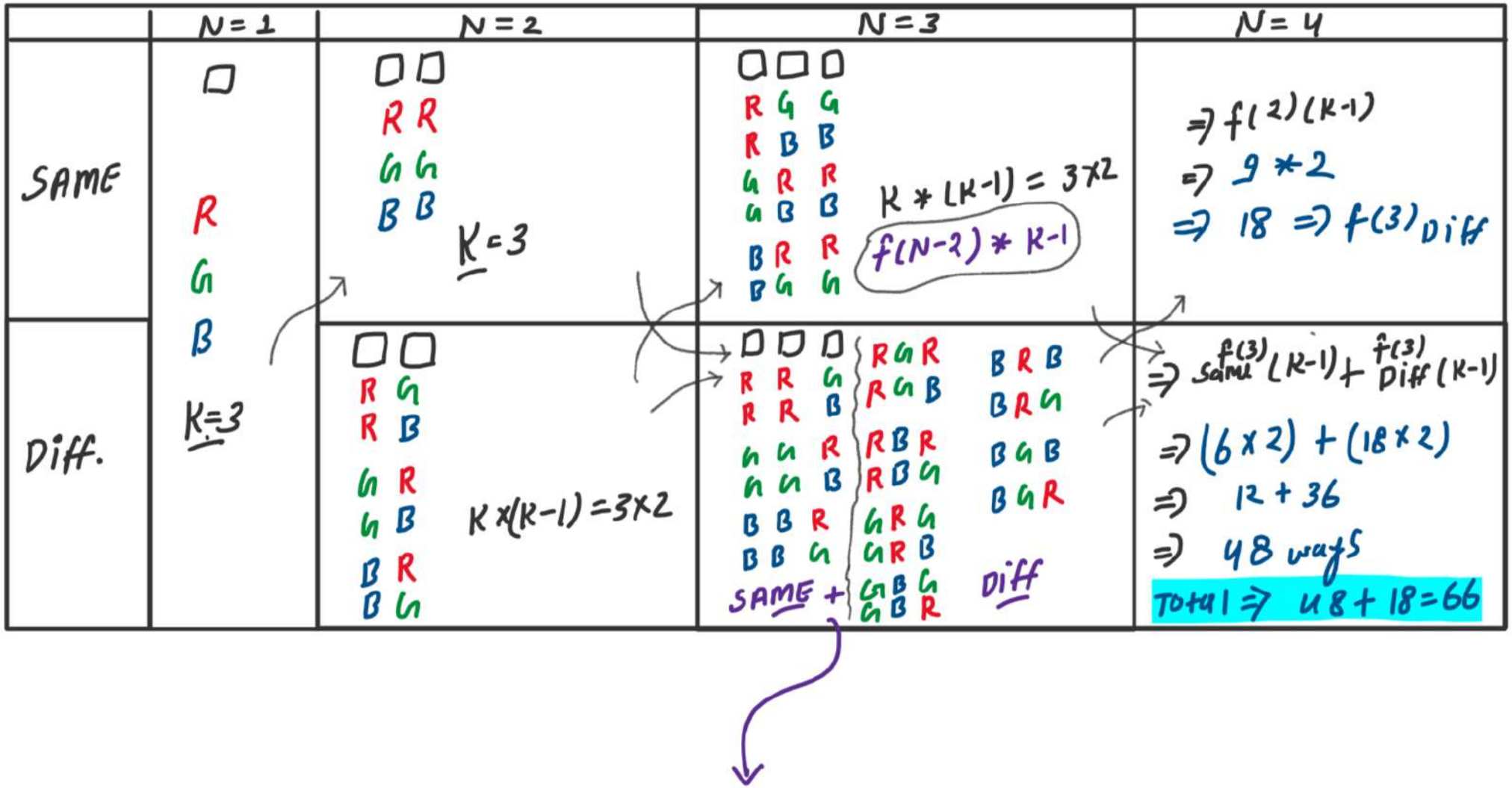
Input : N = 1 K = 3
Output : 3

Input : N = 2 K = 3
Output : 9

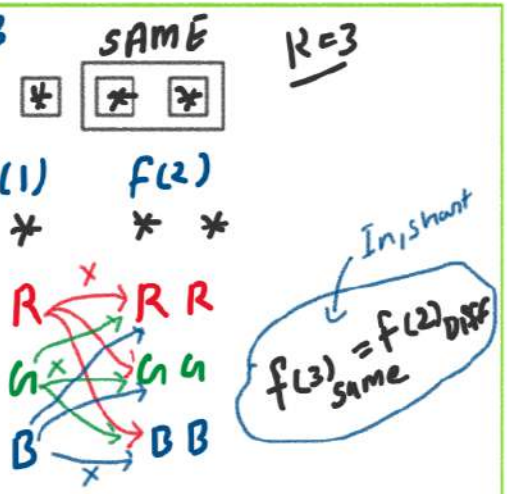
Input : N = 3 K = 3
Output : 24

Input : N = 4 K = 3
Output : 66




N = 1
N = 2
N = 3
N = 4
SAME
Diff.
K=3
K=3
K x (K-1) = 3x2
K * (K-1) = 3x2
f(N-2) * K-1
SAME +
Diff
⇒ f(2)(K-1)
⇒ 9 * 2
⇒ 18 ⇒ f(3) Diff
⇒ f(3) Same (K-1) + f(3) Diff (K-1)
⇒ (6 x 2) + (18 x 2)
⇒ 12 + 36
⇒ 48 ways
Total ⇒ 48 + 18 = 66

N=3 SAME K=3



$f(3)_{SAME} = f(1) * (K-1)$

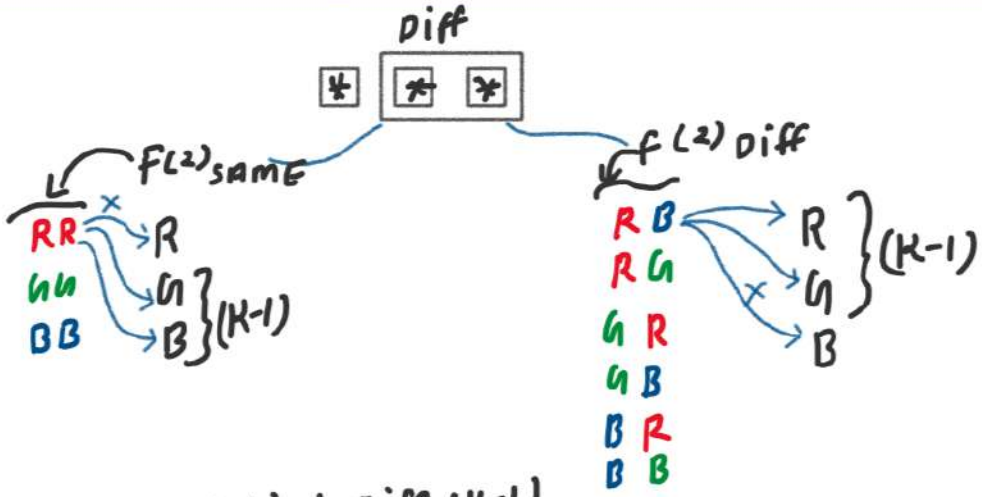
$= f(N-2) * (K-1)$

$f(3)^{same}_{ways} \Rightarrow 3 * 2$

$\Rightarrow 6$ ways.

+

N=3 K=3



$f(3)_{Diff} = same(K-1) + Diff(K-1)$

$= (K-1)[same + Diff]$

$= (K-1)[f(2)]$

$f(3)^{Diff}_{ways} \Rightarrow 2[9]$

$\Rightarrow 18$ ways

$$= \text{Total ways } f(3)$$

$$= f(3)_{same} + f(3)_{diff}$$

$$= [f(n-2) * (k-1)] + [f(n-1) * (k-1)]$$

$$\text{Total ways } f(N) = (k-1) * [f(n-2) + f(n-1)]$$

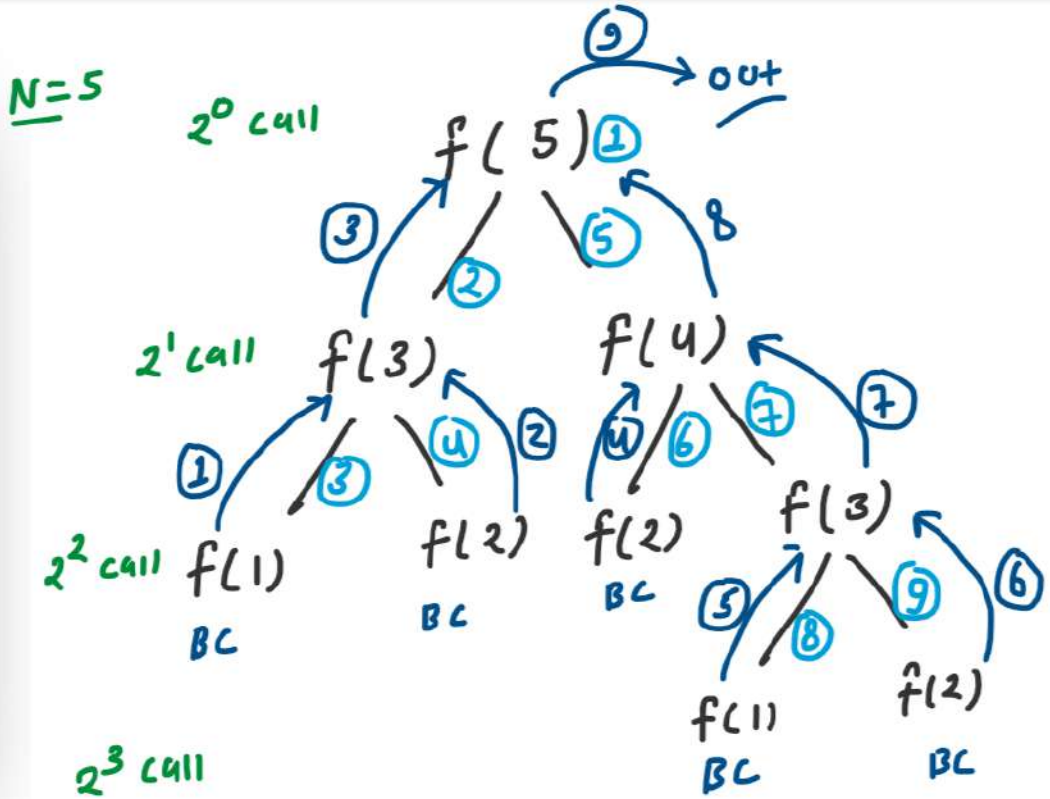
RECURSIVE RELATION

```cpp
#include<iostream>
using namespace std;

int getPaintWays(int N, int K){
    // Base Case
    if(n == 1){
        return K;
    }
    if(n == 2){
        return K + (K*(K-1));
    }

    int ans = (K-1) * (getPaintWays(N-2,K) + getPaintWays(N-1, K));
    return ans;
}

int main(){
    int N = 3; // Posts
    int K = 3; // Colors
    cout<<getPaintWays(N,K)<<endl;
}
```



Total Entry in stack = 5
= N
S.C. = O(N)

T.C. = $O(2^3)$
= $O(2^{N-2})$ ⇒ $O(2^N)$