Counting Configurations

Year 2022– 2023

ALEXANDRA BURBULEA, SORIN-GABRIEL GABĂRĂ, ALEXANDRU IONIȚĂ, MIHAI MAXIM, RĂZVAN TĂNASĂ, ADRIAN SOLCANU, VICTOR SOLCANU (8th grade)

School: Colegiul Naţional *C. Negruzzi*, Iaşi

Teachers: ADRIAN ZANOSCHI, Colegiul Național *C. Negruzzi* and GEORGE-IOAN STOICA, student at l'École Polytechnique Paris

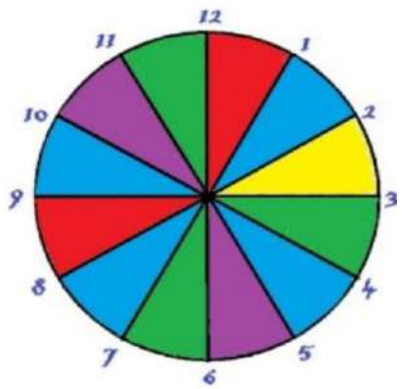
Researcher: IULIAN STOLERIU, (Universitatea *Alexandru Ioan Cuza* din Iaşi)

1. PRESENTATION OF THE RESEARCH TOPIC

The 12 hour sectors of an analogue clock are to be painted in such a way that each sector is painted by one colour and there are no two adjacent sectors having the same colour. If there are 5 colours available, in how many ways can the painting be done? Generalize for n sectors and k available colours.

Imagine now a big circular pizza that is cut into 12 equal slices and 5 available toppings. The pizza must be topped in such a way that each slice has a different topping and there are no two adjacent slices having the same topping. In how many ways can this be done? Generalize to n slices and k toppings.

Problems that require determining the number of configurations of a possible map are essential for combinatorial problems and algorithms for computer programming. We want to present how many combinations of painting the sectors of the circle are possible, if the adjacent colours cannot be the same. We will solve this in the particular case of 12 sectors and 5 colours and then we will find a generalisation. Moreover, we want to find out the number of configurations (which respect the same conditions) when those obtained from each other by rotation are identified.



2. SOLUTION FOR THE FIRST PART

The first task is to find a general formula for ***the number of ways to colour n sectors by k colours without counting for repetitions*** (the configurations that can be obtained by rotation are not identified).

Our first idea was to try and calculate the number of possibilities for smaller and trivial cases.

Also, these are some rules that we shall obtain (for n, the number of sectors, and k, the number of colours):

2. $k \geq 2$, so that the number of configurations is larger than 1;

3. If $k = 2$, then the number of sectors is always even.

Proof:

Let's suppose that $k = 2$ and n is odd:

n odd $\Rightarrow n = 2x + 1, x \in \mathbb{N}^*$

$k = 2$: colour a and colour b

We would have:

Sector 1: colour a

Sector 2: colour b

Sector 3: colour a

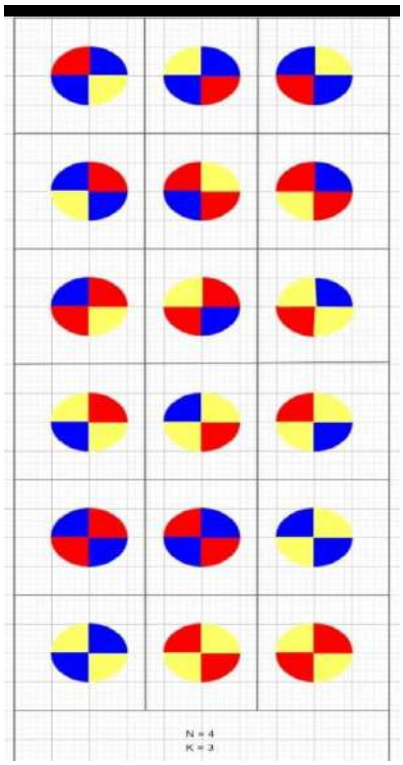
…

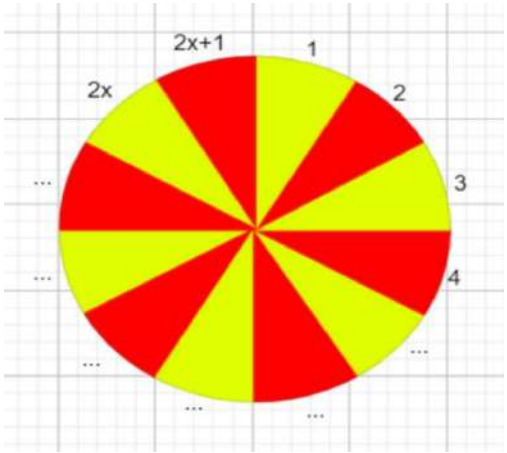
Sector $2x - 1$: colour a

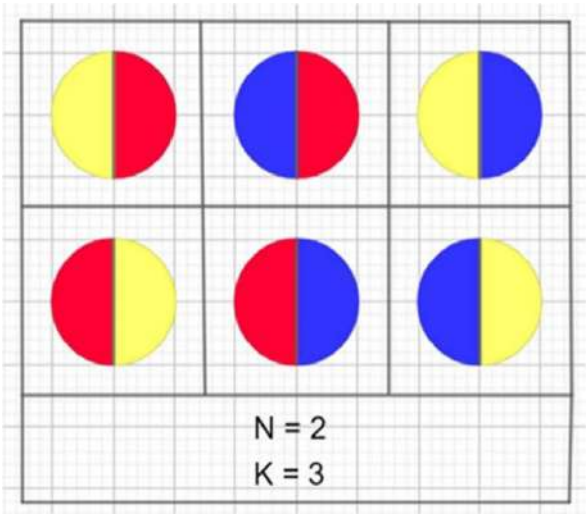
Sector $2x$: colour b

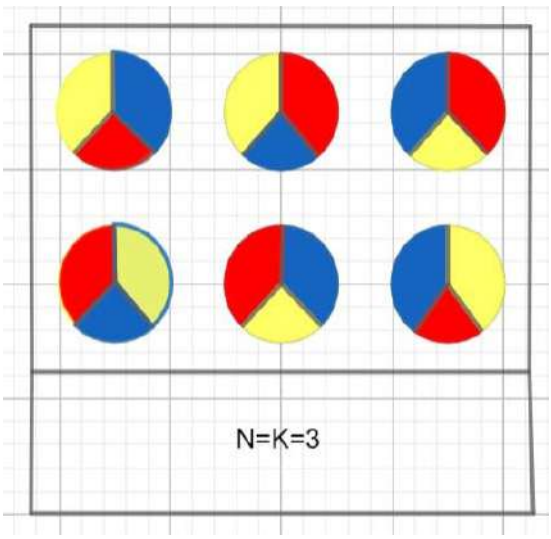
Sector $2x + 1$: colour a

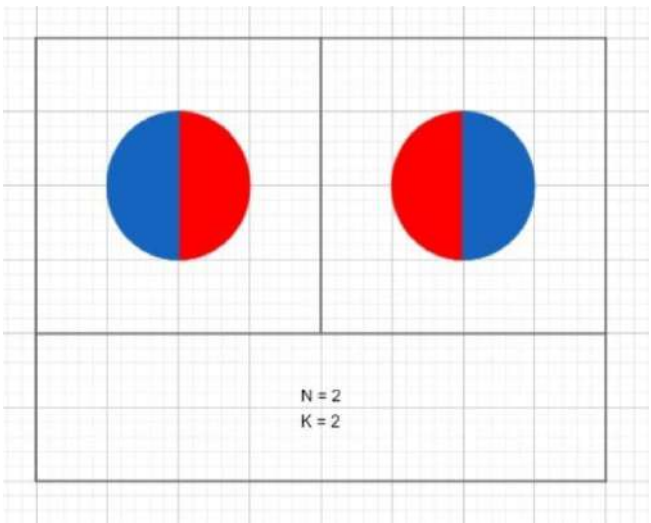
But, sector $2x + 1$ is adjacent to sector 1, so they can't have the same colour. So, if $k = 2, n \neq 2x + 1 \Rightarrow n$-even.




2x+1
1
2x
2
3
4


N = 2
K = 3


N=K=3


N = 2
K = 2

As we have observed in the earlier part of our presentation, we must determine in how many ways can we colour the circle so that no two adjacent sectors have the same colour. So, we will let $a(n,k)$ be the number of configurations for n sectors and k colours (1).

In the case where $n = 12$ and $k = 5$, we may assume that the first sector can be coloured in 5 ways and the other ones in 4 ways, because they cannot get the colour of the previous sector. Hence, the result should be $5 \cdot 4^{11}$. However, this is not the right answer, as we still take in consideration some configurations where the last and first have the same colour.

That's why we thought that we can find a way to subtract those cases. The easiest way to do it is to consider the last and first sector as only one unit. This means we must count the valid configurations for a clock with 11 sectors and 5 colours.

However, this can be written as $a(11,5)$. Actually, you can immediately see the process is repetitive and it goes until you reach $a(1,5)$, which is equal to 0 (because then the first and last sectors never have different colours).

Hence, we have

$$a(12,5) = 5 \cdot 4^{11} - a(11,5)$$
$$a(11,5) = 5 \cdot 4^{10} - a(10,5)$$
$$a(10,5) = 5 \cdot 4^{9} - a(9,5)$$
$$\dots$$
$$a(2,5) = 5 \cdot 4^{1} - a(1,5)$$
$$a(1,5) = 0$$

If we start calculating, it means that:

$$a(2,5) = 5 \cdot 4^1$$
$$a(3,5) = 5 \cdot 4^2 - (5 \cdot 4^1 - 5) = 5 \cdot 4^2 - 5 \cdot 4^1$$
$$a(4,5) = 5 \cdot 4^3 - (5 \cdot 4^2 - 5 \cdot 4^1 + 5) = 5 \cdot 4^3 - 5 \cdot 4^2 + 5 \cdot 4^1$$
$$a(5,5) = 5 \cdot 4^4 - (5 \cdot 4^3 - 5 \cdot 4^2 + 5 \cdot 4^1 - 5) = 5 \cdot 4^4 - 5 \cdot 4^3 + 5 \cdot 4^2 - 5 \cdot 4^1$$

We can easily observe that the signs alternate, depending on the parity of n (even or odd). The number of configurations for the case of $n = 12$ and $k = 5$ will look like this:

$$a(12,5) = 5 \cdot 4^{11} - 5 \cdot 4^{10} + 5 \cdot 4^9 - 5 \cdot 4^8 + \dots + 5 \cdot 4^1$$

If we subtract $k = 5 = 5 \cdot 4^0$, we have

$$a(12,5) - 5 = 5 \cdot 4^{10}(4-1) + 5 \cdot 4^8(4-1) + 5 \cdot 4^6(4-1) + 5 \cdot 4^4(4-1) + 5 \cdot 4^2(4-1) + 5 \cdot 4^0(4-1)$$
$$= 15\,(4^{10} + 4^8 + 4^6 + 4^4 + 4^2 + 1)$$

However, we can see that if try to write a simplified form for a case with an odd number of sectors, we won't need to subtract k anymore. As an example, we can take $a(5,5)$. So, we have:

$$a(5,5) = 5 \cdot 4^4 - (5 \cdot 4^3 - 5 \cdot 4^2 + 5 \cdot 4^1 - 5) = 5 \cdot 4^4 - 5 \cdot 4^3 + 5 \cdot 4^2 - 5 \cdot 4^1$$
$$= 5 \cdot 4^3(4-1) + 5 \cdot 4^1(4-1) = 15 \cdot (4^3 + 4^1) = 15 \cdot 4 \cdot (4^2 + 1)$$

In order to narrow down the number of configurations even further, we can take another two subcases: one for an even value of n, and another one for an odd value of n.

Case I: n even $(n = 2x)$

We get

$$a(2x,k)$$
$$= (k-2)\cdot k\cdot\left[(k-1)^{2x-2}+(k-1)^{2x-4}+\ldots+(k-1)^{2x-(2x-2)}+(k-1)^0\right]+k$$
$$= (k-2)\cdot k\cdot\{[(k-1)^2]^{x-1}+[(k-1)^2]^{x-2}+\ldots+[(k-1)^2]+[(k-1)^2]^0\}+k$$
$$= (k-2)\cdot k\cdot\frac{[(k-1)^2]^x-1}{(k-1)^2-1}+k$$
$$= (k-2)\cdot k\cdot\frac{[(k-1)^2]^x-1}{(k-1-1)(k-1+1)}+k$$
$$= (k-2)\cdot k\cdot\frac{[(k-1)^2]^{\frac{n}{2}}-1}{(k-2)\cdot k}+k$$
$$= [(k-1)^2]^{\frac{n}{2}}-1+k=(k-1)^n-1+k$$

Case II: n odd $(n = 2x+1)$

We get

$$a(2x+1,k)$$
$$= (k-2)\cdot k\cdot[(k-1)^{2x+1-2}+(k-1)^{2x+1-4}+\cdots+(k-1)^3+(k-1)^1]=$$
$$(k-2)\cdot k\cdot(k-1)\cdot[(k-1)^{2x-2}+(k-1)^{2x-4}+\cdots+(k-1)^2+(k-1)^0]$$
$$= (k-2)\cdot k\cdot(k-1)\cdot\{[(k-1)^2]^{x-1}+[(k-1)^2]^{x-2}+\cdots+[(k-1)^2]^1+$$
$$+[(k-1)^2]^0\}$$
$$= (k-2)\cdot k\cdot(k-1)\cdot\frac{[(k-1)^2]^x-1}{(k-1)^2-1}$$
$$= (k-2)\cdot k\cdot(k-1)\cdot\frac{[(k-1)^2]^x-1}{(k-1-1)\cdot(k-1+1)}$$
$$= (k-2)\cdot k\cdot(k-1)\cdot\frac{[(k-1)^2]^x-1}{(k-2)\cdot k}$$
$$= (k-1)\cdot\{[(k-1)^2]^x-1\}$$
$$= (k-1)\cdot\left\{[(k-1)^2]^{\frac{n-1}{2}}-1\right\}$$
$$= (k-1)\cdot[(k-1)^{n-1}-1]$$
$$= (k-1)\cdot(k-1)^{n-1}-(k-1)$$
$$= (k-1)^{1+n-1}-k+1=(k-1)^n-k+1$$

In conclusion, the formulae that we have obtained are:

I. $(k-1)^n-k+1$, if n is odd.

II. $(k-1)^n-1+k$, if n is even.

The general formula for both cases (n even or n odd) can be written as: (2)

$$\boldsymbol{a(n,k)=(k-1)^n+(-1)^n\cdot(k-1)}$$

In particular, for $n = 12$ and $k = 5$, we get $a(12,5) = 4^{12}+4 = 16777220$ configurations and for $n = 12$ and k =8, we get $a(12,8) = 13841287208$ configurations.

```
from operator import eq
from itertools import product


def anyeq(a):
    a.append(a[0])
    return not any(map(eq, a, a[1:]))


colors = [1, 2, 3, 4, 5]

combi = product(colors, repeat=12)
print(len([1 for x in combi if anyeq(list(x))]))
```

This was the first code we have written. Although it is very slow, because it counts the configurations one at a time, it helped us find the values for smaller numbers, before finding the formulas. Basically, as we defined the function anyeq, we check if a distribution is or not possible, just by bashing. Then, we count of all valid configurations and obtain the result (3). Thus, this algorithm searches for each possible combination of colours, so this is not very efficient, but it confer a correct result.

Now, we are going to move on to the C++ solution for the first part of the problem. This code was created by converting the formulas obtained above into the C++ programming language.

The C++ code helps us

1) find values of higher pairs of numbers;
2) reduce the time in which the solutions can be found;
3) reduce the chance of an error occurring.

This is the C++ code for the first part of the problem. We chose a set of variables to help us find the solution: k – the number of colours, n – the number of sectors in which the analogue clock will be divided. The code only works if the number of colours and the numbers of sectors are larger than 1.

```cpp
#include <iostream>

using namespace std;

int main()
{
    int k,n,i,x;
    cout<<"Introduce the number of sectors: ";
    cin>>n;
    cout<<"Introduce the number of colours: ";
    cin>>k;
    x=1;
    if(n%2==0)
    {
        for(i=0;i<n;i++)
        {
            x=x*(k-1);
        }
        cout<<"The number of solutions is: "<<x-1+k;
    }
    else
    {
        for(i=0;i<n;i++)
        {
            x=x*(k-1);
        }
        cout<<"The number of solutions is: "<<x+1-k;
    }
    return 0;
}
```

We chose to divide the problem into two separate parts:

I. When n is divisible by 2 (even)

II. When the remainder of n divided by 2 is 1 (odd)

I. When n is divisible by 2 (even, $2|n$).

```cpp
if(n%2==0)
{
    for(i=0;i<n;i++)
    {
        x=x*(k-1);
    }
    cout<<"The number of solutions is: "<<x-1+k;
}
```

Within these lines we want to reach the formula:

$$(k-1)^n + k - 1$$

To do that, we need to choose a variable x, which is initially equal to 1, and another variable i that starts at 0 and will increase by 1, until it reaches n. Every time i increases, x is multiplied by $(k-1)$, which is the correspondent of the line $x = x \cdot (k-1)$.

II. When is not divisible by 2 (n is odd)

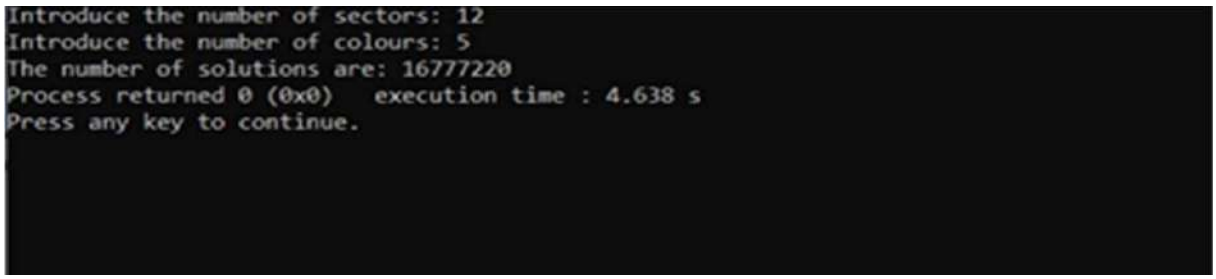
```
else
{
    for(i=0;i<n;i++)
    {
        x=x*(k-1);
    }
    cout<<"The number of solutions are: "<<x+1-k;
```

Within these lines we want to reach the formula:

$$(k-1)^n - k + 1$$

To do that, we need to choose a variable x, which is initially equal to 1, and another variable i that starts at 0 and will increase by 1, until it reaches n. Every time i increases, x is multiplied by $(k-1)$, which is the correspondent of the line $x = x \cdot (k-1)$;

Finally, outputs computed by the code are shown in a tab looking like this:



I.	$n = k = 2$	2 possibilities
II.	$n = 2, k = 3$	6 possibilities
III.	$n = k = 3$	6 possibilities
IV.	$n = 4, k = 3$	18 possibilities
V.	$n = 5, k = 3$	30 possibilities
VI.	$n = 5, k = 4$	240 possibilities
VII.	$n = 5, k = 5$	1020 possibilities
VIII.	$n = 6, k = 3$	66 possibilities
IX.	$n = 6, k = 4$	732 possibilities
X.	$n = 6, k = 5$	4100 possibilities
XI.	$n = k = 6$	15630 possibilities
XII.	$n = 8, k = 3$	258 possibilities
XIII.	$n = 8, k =4$	6564 possibilities
XIV.	$n = 9, k = 3$	510 possibilities
XV.	$n = 9, k = 4$	19680 possibilities
XVI.	$n = 9, k = 5$	262140 possibilities
XVII.	$n = 9, k = 6$	1953120 possibilities

3. SECOND PART

In the second part of the problem, we need to focus on ***the number of configurations when those obtained from each other by rotation are identified***. Although in this section, we didn't conclude with a formula or a result, we think that our idea can result into a possible solution.

Intuitively, we can start by considering all the configurations, and then eliminate one by one the ones which repeat with respect to rotation. We observe that for a configuration in which all colours are different there are exactly n possibilities of permutations, so we count just 1 with respect to rotation. Now, as a function of the number of colours and their positioning, we observe that there can be computed a particular number of repetitions and we can divide all correct ones to the number of repetitions.

We are not sure yet, but we think that we can derive a formula based on these observations and inclusion-exclusion principle.

We will finish by considering a particular case that may be generalized later. For the particular case $n = 6, k = 4$, we computed the number of configurations as it follows.

We assume that the four colours are: A, B, C and D.

- **If two colours are used,** we will have 6 possibilities (e.g. 1 possible arrangement (4) is ABABAB), which are the 6 different groups of two colours (AB, AC, AD, BC, BD, CD)).
- **If three colours are used,** we will have:

  I. Using 1A, 2B, 3C , the only possible arrangement is: CBCBCA.
  4 (ways to choose the 3 colours) $\times$ 6 (permutations)= 24 possibilities

  II. Using 2A, 2B, 2C, the possible arrangements are: ABCABC, ABACBC, ABCBAC, ABCACB and ACBACB.
  4 (ways to choose the 3 colours) $\times$ 5 (arrangements)=20 possibilities

  Total (for three colours used): $24 + 20 = 44$ possibilities.

- **If four colours are used,** we will have:

  I. Using 1A, 1B, 1C, 3D, the possible arrangements are: ADBDCD and ADCDBD.
  2 (arrangements) $\times$ 4 (permutations) = 8 possibilities

  II. Using 1A, 1B, 2C, 2D, we can choose in six ways which colours to use twice.
  $\frac{5!}{1!\,2!\,2!} = 30$ ways to arrange ABCCDD (5), including:
  $\frac{4!}{1!1!2!} \cdot 2 = 24$ cases when two sectors coloured in C or D are adjacent
  $\frac{3!}{1!1!1!} = 6$ cases when two sectors coloured in C are adjacent and two sectors coloured in D are adjacent.

  The total number of configurations when four colours are used is
  $$8 + 6 \cdot (30 - 24 + 6) = 8 + 6 \cdot 12 = 80 \text{ possibilities}$$

Total = $6 + 44 + 80 = 130$ possibilities.

4. CONCLUSION

To solve this problem, we only needed to use calculus. In order to find the formulas that we used to generalize the problem for *n* sectors and *k* colours, we analysed a few particular cases to grasp an idea about how to approach the problem. Also, we have written a C++ code to determine the solutions for higher cases.

To solve the second part of the problem, we tried to find a formula based on the number of divisors of n. Also, we calculated a particular case to verify the formula.

To reach the final formula, we used the results found during the solution.

Editing Notes

(1) So, $a(n,k)$ is the number of sequences of n colours taken among k possible colours, where two consecutive colours are different and the last is different from the first.

(2) In fact, in both cases even or odd n, without grouping the terms two by two, we have $a(n,k) = k\cdot(k-1)^{n-1} - k\cdot(k-1)^{n-1} + \cdots + (-1)^{n-1}k\cdot(k-1)^2 + (-1)^n k\cdot(k-1)$, which is the sum of a geometric progression of ratio – $(k-1)$ and can be calculated without separating the cases.

(3) More precisely, the program constructs a list of all the sequences of 12 colours taken from those of colors (combi=product(colors, repeat=12)); anyeq(a) compares a sequence a with itself shifted by one rank, builds a list of their coincidences and returns true if there are none, i.e. if it is a valid configuration. Finally, len[1 for x in combi if anyeq(x)] builds a list consisting of an element "1" for each valid configuration and yields the number of elements in this list, that is the number of valid configurations.

(4) Here and in the following, "possible arrangement" has to be understood after identifying those obtained by rotation ; e.g. with only two colours A and B, the possible configurations before identification are ABABABAB and BABABA, which are counted as 1 arrangement only.

(5) Here, the calculation is done differently. Firstly, all the 6-colour sequences modulo the rotations that have the distribution ABCCDD are counted: with a possible rotation, we can assume that A is the colour of the first sector (in general, we can take the first of the two isolated colours). So we obtain the number of ways to choose a B, 2 C and 2 D for the other 5 sectors, which is the multinomial coefficient $5!/(1!\,2!\,2!)$.

Next, we have to subtract the number of configurations that are not valid. By collapsing CC, the number of sequences (beginning with A) in which C appears in two consecutive sectors is the same as the number of 4 colour-sequences having distribution BCDD, so by the same calculation we have $4!/(1!\,1!\,2!)$ arrangements of this type.

Now, after subtracting this number for sequences containing CC and for those containing DD, we have subtracted twice those containing both. By the same argument, there are $3!/(1!\,1!\,1!)$ such sequences and this number has to be added.

So we find the result given for the number of valid configurations with distribution ABCCDD modulo rotations: $\frac{5!}{(1!2!2!)} - \frac{4!}{(1!1!2!)} + \frac{3!}{(1!1!1!)} = 30 - 24 + 6.$