*M.A.Rodichev*

**Optimisation of algorithms in programming by means of problem analysis using the example of the magic square**

T

*In this material the author demonstrates how a change of approach to a given mathematical problem contributes to its fastest solution.*

**Introduction**

his paper would not have been possible without mathematicians who have a thirst for beauty in their understanding. One of these beauties is the magic square, familiar to many from school years, an n x n square in which the numbers do not repeat and their sums in columns and rows are equal.  
Such problems are given to children in the second grade, in order to develop their thinking, but, as you understand, it is too simple a problem, and it was complicated by adding the condition that the numbers are squared.  
I encountered one of these squares in Numberphile. Then I had the idea to find more such squares, using a computer.  
  
**Method of solving these problems head on**

The good thing about this method is that you don't have to think. It was suggested by the vast majority of people in the comments under Numberphile's video about the square. Its essence is just going through all values and waiting when the given problem conditions are fulfilled. After all, we don't do the math, the machine does, so we can ignore that it takes about 100^9 calculations, even using parallel computing we get roughly 5 000 days to try all the options for the magic 3x3 square, which is a pretty long time (the required number of days is based on the processing power of the Ryzen 5 5300U processor). This number of days makes you wonder how we could simplify this task to make the search faster since we used all the technique possibilities (parallel search).

**Consistent analysis of the problem**

Nine numbers arranged in a certain order in rows and columns? Yes, it is, but if you take into account that each row and column gives the same sum when you squared their numbers, then you can come to the conclusion that the magic square is not just 9 numbers, it is 6 triples of numbers (3 rows, 3 columns) giving the same value. So, we need to find such a sum that can be "coded" by 6 or more triples of numbers instead of just going through all 9 "positions" of the magic square. What does this mean for us? It means that we only have to search through 3 positions and "write" those triples that give the same sums, that is, we get 100^3 calculations, which is a much smaller value.  
Everything seems fine, but let's not forget one more rule: "Numbers do not repeat", and in our search it is possible:

62 + 622 + 872 = 11 449  
622 + 62 + 872 = 11 449  
872 + 622 + 62 = 11 449

And so on, I think being familiar with combinatorics you will notice that this would be 3! repetitions, but what does this mean for us?   
First of all, these repetitions increase the number of calculations and also create an obstacle in the form of another 5 lines that do not advance our solution in any way.  
To find a pattern and figure out how to get rid of unnecessary operations, I had to start adding up the squares of numbers myself, within single digits. Having done this work, I came to the conclusion that the enumeration of the numbers where:

1st number: a  
2nd number: a+1  
3rd number: a+1+1

Will always give different sequences of numbers, and will not give repeating strings, where only replaced the numbers in places, so instead of trying 6 options, we find 1 and stop, and satisfy the condition of our problem, every number is not repeatable.  
So, from the brute force problem analysis we came up with a more elegant solution.  
Initially we had 100^9 calculations, then using analysis we obtained 100^3 calculations. Remembering the condition that numbers do not repeat, we achieved the minimum value of 156,849 calculations.  
To summarize, out of 1,000,000,000,000,000,000,000 calculations we came to 156,849.  
An outstanding result, so far this is the maximum simplification, which I was able to come to, the next piece of work will be devoted to solving the problem of machine understanding the square, because at this stage, it is just a string, which can not always form our square.   
  
**The problem of machine comprehension**

We have learned to find strings of 3 numbers that satisfy our conditions, but how do we teach the computer to understand if it is possible to assemble a square from the data?  
To solve this, we will need to use the method we once used - solving by ourselves and writing out the square line by line, we can start noticing that each number occurs twice, i.e. at the intersection of the row and column.  
And this is true, by conditioning the computer that each number of a row occurs twice, in that row itself and in another of the rows, we will only get rows in which the values are related to other rows, and so can form a square.  
This is somewhat similar to Eratosthenes' lattice principle, we will also remove rows of 3 numbers if at least one of them is not repeated twice, leaving only those that are repeated two or more times.

**Conclusion**

In conclusion I would like to tell that without consecutive analysis of the set problem, there is a great chance to choose a wrong way of its decision, the most primitive way of the analysis is the decision on a paper, when we start to solve a problem independently, unconsciously or consciously there is a desire to search laws, not to go through all manually, otherwise, starting directly to the computer, it is possible to be stuck on opinion "Machines we have powerful, will quickly calculate", but it, unfortunately, not so. Next, in "result of algorithm's work", I will demonstrate all magic squares I found.

**The result of the algorithm.**

The results are amazing, there is almost no delay in getting an answer, which is great, and we ended up finding 20(25) squares:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 02 | 252 | 602 |  | 0 | 625 | 3600 | 4225 |
| 392 | 482 | 202 |  | 1521 | 2304 | 400 | 4225 |
| 522 | 362 | 152 |  | 2704 | 1296 | 225 | 4225 |
|  |  |  |  | 4225 | 4225 | 4225 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 02 | 402 | 752 |  | 0 | 1600 | 5625 | 7225 |
| 512 | 602 | 322 |  | 2601 | 3600 | 1024 | 7225 |
| 682 | 452 | 242 |  | 4624 | 2025 | 576 | 7225 |
|  |  |  |  | 7225 | 7225 | 7225 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 52 | 542 | 782 |  | 25 | 2916 | 6084 | 9025 |
| 302 | 752 | 502 |  | 900 | 5625 | 2500 | 9025 |
| 902 | 222 | 212 |  | 8100 | 484 | 441 | 9025 |
|  |  |  |  | 9025 | 9025 | 9025 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 92 | 462 | 782 |  | 81 | 2116 | 6084 | 8281 |
| 622 | 542 | 392 |  | 3844 | 2916 | 1521 | 8281 |
| 662 | 572 | 262 |  | 4356 | 3249 | 676 | 8281 |
|  |  |  |  | 8281 | 8281 | 8281 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 82 | 512 | 962 |  | 64 | 2601 | 9216 | 11881 |
| 692 | 722 | 442 |  | 4761 | 5184 | 1936 | 11881 |
| 842 | 642 | 272 |  | 7056 | 4096 | 729 | 11881 |
|  |  |  |  | 11881 | 11881 | 11881 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 12 | 382 | 862 |  | 1 | 1444 | 7396 | 8841 |
| 462 | 792 | 222 |  | 2116 | 6241 | 484 | 8841 |
| 822 | 342 | 312 |  | 6724 | 1156 | 961 | 8841 |
|  |  |  |  | 8841 | 8841 | 8841 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 22 | 312 | 94 |  | 4 | 961 | 8836 | 9801 |
| 492 | 822 | 262 |  | 2401 | 6724 | 676 | 9801 |
| 862 | 462 | 172 |  | 7396 | 2116 | 289 | 9801 |
|  |  |  |  | 9801 | 9801 | 9801 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 582 | 142 | 792 |  | 3364 | 196 | 6241 | 9801 |
| 312 | 862 | 382 |  | 961 | 7396 | 1444 | 9801 |
| 742 | 472 | 462 |  | 5476 | 2209 | 2116 | 9801 |
|  |  |  |  | 9801 | 9801 | 9801 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 112 | 232 | 712 |  | 121 | 529 | 5041 | 5691 |
| 432 | 592 | 192 |  | 1849 | 3481 | 361 | 5691 |
| 612 | 412 | 172 |  | 3721 | 1681 | 289 | 5691 |
|  |  |  |  | 5691 | 5691 | 5691 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 12 | 282 | 762 |  | 1 | 784 | 5776 | 6561 |
| 442 | 642 | 232 |  | 1936 | 4096 | 529 | 6561 |
| 682 | 412 | 162 |  | 4624 | 1681 | 256 | 6561 |
|  |  |  |  | 6561 | 6561 | 6561 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 22 | 432 | 742 |  | 4 | 1849 | 5476 | 7329 |
| 592 | 582 | 222 |  | 3481 | 3364 | 484 | 7329 |
| 622 | 462 | 372 |  | 3844 | 2116 | 1369 | 7329 |
|  |  |  |  | 7329 | 7329 | 7329 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 832 | 432 | 592 |  | 6889 | 1849 | 3481 | 12219 |
| 732 | 312 | 772 |  | 5329 | 961 | 5929 | 12219 |
| 12 | 97 | 532 |  | 1 | 9409 | 2809 | 12219 |
|  |  |  |  | 12219 | 12219 | 12219 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 22 | 342 | 532 |  | 4 | 1156 | 2809 | 3969 |
| 432 | 382 | 262 |  | 1849 | 1444 | 676 | 3969 |
| 462 | 372 | 222 |  | 2116 | 1369 | 484 | 3969 |
|  |  |  |  | 3969 | 3969 | 3969 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 42 | 232 | 522 |  | 16 | 529 | 2704 | 3249 |
| 322 | 442 | 172 |  | 1024 | 1936 | 289 | 3249 |
| 472 | 282 | 162 |  | 2209 | 784 | 256 | 3249 |
|  |  |  |  | 3249 | 3249 | 3249 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 32 | 262 | 662 |  | 9 | 676 | 4356 | 5041 |
| 462 | 512 | 182 |  | 2116 | 2601 | 324 | 5041 |
| 542 | 422 | 192 |  | 2916 | 1764 | 361 | 5041 |
|  |  |  |  | 5041 | 5041 | 5041 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 82 | 642 | 672 |  | 64 | 4096 | 4489 | 8649 |
| 532 | 522 | 562 |  | 2809 | 2704 | 3136 | 8649 |
| 762 | 432 | 322 |  | 5776 | 1849 | 1024 | 8649 |
|  |  |  |  | 8649 | 8649 | 8649 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 292 | 472 | 882 |  | 841 | 2209 | 7744 | 10794 |
| 682 | 672 | 412 |  | 4624 | 4489 | 1681 | 10794 |
| 732 | 642 | 372 |  | 5329 | 4096 | 1369 | 10794 |
|  |  |  |  | 10794 | 10794 | 10794 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 72 | 412 | 912 |  | 49 | 1681 | 8281 | 10011 |
| 612 | 772 | 192 |  | 3721 | 5929 | 361 | 10011 |
| 792 | 492 | 372 |  | 6241 | 2401 | 1369 | 10011 |
|  |  |  |  | 10011 | 10011 | 10011 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 62 | 622 | 872 |  | 36 | 3844 | 7569 | 11449 |
| 732 | 662 | 422 |  | 5329 | 4356 | 1764 | 11449 |
| 782 | 572 | 462 |  | 6084 | 3249 | 2116 | 11449 |
|  |  |  |  | 11449 | 11449 | 11449 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 62 | 272 | 742 |  | 36 | 729 | 5476 | 6241 |
| 382 | 662 | 212 |  | 1444 | 4356 | 441 | 6241 |
| 692 | 342 | 182 |  | 4761 | 1156 | 324 | 6241 |
|  |  |  |  | 6241 | 6241 | 6241 |  |

**Errors in the algorithm.**

Nothing is perfect and even this well thought out algorithm has produced an error in the subsequent sums not being collected in the magic square, but he who does not seek will not find, try and collect them yourself.

*Sum = 10150*

6 33 95

6 67 75

15 30 95

15 58 81

30 67 69

33 50 81

45 50 75

45 58 69

*Sum = 5886*

2 29 71

2 49 59

7 19 74

7 46 61

11 17 74

11 49 58

17 46 59

19 22 71

22 41 61

29 41 58

*Sum = 5574*

2 23 71

2 43 61

7 22 71

7 41 62

19 37 62

19 43 58

22 37 61

23 41 58

*Sum = 7461*

1 8 86

1 58 64

4 7 86

4 46 73

7 44 74

8 34 79

14 32 79

14 44 73

31 32 74

31 56 58

34 47 64

46 47 56

*Sum = 9246*

7 19 94

7 61 74

11 17 94

11 49 82

17 59 74

19 62 71

29 41 82

29 58 71

41 61 62

49 58 59