Train Game Diary

Brute force vs Graph approach

29/12/15

Brute Force

Currently the program uses a brute force approach to find possible solutions. Consider the digits $\{1, 2, 3, 4\}$ and the set of operations $\{+, -, \times, \div\}$. First, all possible permutations of the operations are generated.

$+ + +$	$+ - -$	$+ \times \div$	$- - \div$	$\times \times \times$
$+ + -$	$+ - \times$	$+ \div \div$	$- \times \times$	$\times \times \div$
$+ + \times$	$+ - \div$	$- - -$	$- \times \div$	$\times \div \div$
$+ + \div$	$+ \times \times$	$- - \times$	$- \div \div$	$\div \div \div$

Next, all the possible permutations of the digits are generated.

1234	2134	3124	4123
1243	2143	3142	4132
1324	2314	3214	4213
1342	2341	3241	4231
1423	2413	3412	4312
1432	2431	3421	4321

For each combination of operations, each digit set is applied.

For example, for the operation $+ + +$ the digits 1234, 1243, ... 4321 are applied.

$$1+2+3+4$$
$$1+2+4+3$$
$$1+3+2+4$$
$$...$$
$$4+2+3+1$$
$$4+3+1+2$$
$$4+3+2+1$$

Similarly, for the operation $++-$ the numbers 1234, 1243, ... 4321 are applied.

$$1+2+3-4$$
$$1+2+4-3$$
$$1+3+2-4$$
$$...$$
$$4+2+3-1$$
$$4+3+1-2$$
$$4+3+2-1$$

So on and so forth.

We use `eval` to evaluate each of these statements and let a solution be any combination of the digits and operations that equals 10.

In our example there are 32 solutions.

$$1+2+3+4=10$$
$$1+2+4+3=10$$
$$...$$
$$4+2*3/1=10$$
$$4+3*2/1=10$$

Brute Force Calculations

Definitions

Let set of digits to use $D = \{d_1,\ d_2,\ d_3,\ d_4\ .\ .\ .\ d_m\}$
Let set of operations to use $O = \{o_1,\ o_2,\ o_3,\ o_4\ .\ .\ .\ o_n\}$ where $m, n \in \mathbb{Z}$

$|D| = m =$ number of digits to use.
$|O| = n =$ number of operations you have at your disposal.

Valid statement

A valid statement that can be evaluated is

$$d_{p_1} \quad o_{q_1} \quad d_{p_2} \quad o_{q_2} \quad d_{p_3} \quad o_{q_3} \quad d_{p_4}\ .\ .\ .\ o_{q_n} \quad d_{p_m}$$

where $p, q \in \mathbb{Z}$ such that $1 \leq p \leq m$ and $1 \leq q \leq n$.

Generating Permutations of Numbers

$|D| = m$ (from above).

Out of m numbers, we are using all m digits to generate a valid statement.

m numbers to use, permute m times, ${}^mP_m$.

$${}^mP_m = \frac{m!}{(m-m)!} = \frac{m!}{(0)!} = \frac{m!}{1} = m!$$

Generating Combinations of Operations (with Repetition)

$|O| = n$ (from above).

Out of n operations, we are choosing $(n-1)$ operations to generate a valid statement.

n operations choose $n-1$ (with repetition), $\binom{n+(n-1)-1}{(n-1)}$.

$$\frac{(n+(n-1)-1)!}{(n-1)!(n-1)!} = \frac{(2n-2)!}{(n-1)!^2}$$

Putting it together

As mentioned earlier,

> *For each combination of operations, each digit set is applied.*

Therefore, in a brute force approach, we generate a total of

$$\boxed{\frac{(2n-2)!}{(n-1)!^2} \times m!}$$

statements to be evaluated.

An Application - Our Example

Let set of digits to use $D = \{1, 2, 3, 4\}$
Let set of operations to use $O = \{+, -, \times, \div\}$

$|D| = m = 4$ digits to use
$|O| = n = 4$ operations at disposal

Number of brute force statements to evaluate

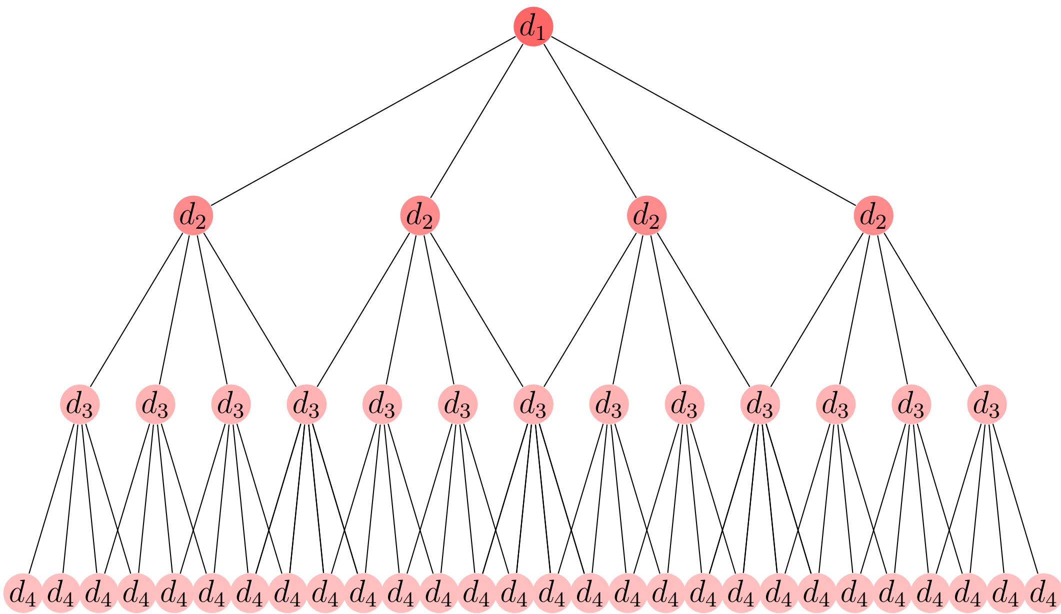
$$\frac{(2n-2)!}{(n-1)!^2} \times m! = \frac{(2(4)-2)!}{((4)-1)!^2} \cdot (4)! = \frac{6!}{(3)!^2} \cdot 4! = 480$$

Complexity

Algorithm time complexity $O(a^2)$.

Graph

Instead, we can approach the problem from a tree-based problem.



Start with a parent node (a digit) say d_1. Then, there are n edges (operations) it can travel down to reach the second node (another digit) d_2. Then there are once again n edges (operations) it can travel down to reach the third node (some other digit) d_3. So on so forth.

To find a solution, we traverse every path, accumulating a result along the way. The end result is compared to the goal required to reach.

Graph Calculations

Single Tree

One tree's depth $= m$ digits deep.

At each level of this tree, there are n possible paths to take.

Total paths to traverse

$$\sum_{i=1}^{m} n^i$$

in one tree

Generalised Trees

Generate a tree for every possible permutation of the digits (without repetition), ${}^mP_m = m!$ trees to traverse.

Putting it all together

Therefore, in a graph approach, we generate a total of

$$\boxed{m! \sum_{i=1}^{m} n^i}$$

statements to evaluated.

An Application - Our Example

Let set of digits to use $D = \{1, 2, 3, 4\}$
Let set of operations to use $O = \{+, -, \times, \div\}$

$|D| = m = 4$ digits to use
$|O| = n = 4$ operations at disposal

Number of brute force statements to evaluate

$$m! \sum_{i=1}^{m} n^i = 4! \sum_{i=1}^{4} 4^i = 24(4 + 4^2 + 4^3) = 2016$$