Article
Danny Wijnschenk · Nov 19, 2017 3m read

Advent of Code 2016 Day19: An Elephant Named Joseph

This is a series of programming challenges for beginners and experienced Caché programmers.

For an introduction : go to article https://community.intersystems.com/post/advent-code-2016-day1-no-time-ta...

Today's challenge is a variation on the White Elephant gift exchange (https://en.wikipedia.org/wiki/Whiteelephantgiftexchange), in this case, by a bunch of Elves where only one can have all gifts.

Each Elf brings a present. They all sit in a circle, numbered starting with position 1. Then, starting with the first Elf, they take turns stealing all the presents from the Elf to their left. An Elf with no presents is removed from the circle and does not take turns.

With 5 elves, the Elf that sits in position 3 gets all the presents.

You will have to calculate the same, but with a bigger number of Elves (look for a full explanation to http://adventofcode.com/2016/day/19, it contains also the puzzle input (number of elves) to start).

I had more that 3 million elves in my party. I decided to arrange all of them in an array, and in a loop, I used a $Order to get to the next elf, and killed that one (the node, not the elf). The loop stops when only one (lucky) Elf is left.

Here is the code for the first part that Bert wrote, he uses the same strategy :

```
Start() PUBLIC {
  s elfsCount=3004953
  for i=1:1:elfsCount {
    s elfs(i)=""
  }
  s elfs=elfsCount

  s currentElf=""
  while elfs>1 {
    s nextElf=$$getNextElf(.elfs,currentElf)
    s takesFromElf=$$getNextElf(.elfs,nextElf)
    s currentElf=takesFromElf
    k elfs(takesFromElf)
    s elfs=elfs-1
  }
  w !
  zw elfs
}

getNextElf(Elfs,CurrentElf) {
  s nextElf=$ORDER(Elfs(CurrentElf))
  if nextElf="" {
    s nextElf=$ORDER(Elfs(""))
  }
  q nextElf
}
```

The second part of the challenge changed the game : instead of stealing from the next Elf, you need to steal from the Elf sitting opposite.

I first thought on not changing my code too much and kept $ordering until I was halfway, but that would be to slow given the heap of Elves.

My second thought was to change the array into a string, or an array of strings to do less $ordering, but that was also much too slow.

Finally, the light went on, and I realized I would only have to calculate the position of the opposite Elf once (at the beginning : count/2), and for all further loops, I could move the opposite Elf one or two positions further, depending on odd/even Elves left.

Here is the code i wrote for the second part :

```
ClassMethod Part2()
{
  #dim elves as %Integer = 3005290
  For elve=1:1:elves {
    Set elves(elve)=1
  }
  Set count=elves
  set elve=""
  Set tries=$Select(count#2:(count-1)/2,1:(count/2))
  Set elveNext=tries+1
  For {
    Set elve=$Order(elves(elve))
    If elve="" Set elve=$Order(elves(""))
    kill elves(elveNext)
    Set elveNext=$Order(elves(elveNext))
    If elveNext="" Set elveNext=$Order(elves(""))
    If count#2 {
      Set elveNext=$Order(elves(elveNext))
      If elveNext="" Set elveNext=$Order(elves(""))
    }
    Set count=count-1
    If count=1 Quit
  }
  Write "elve left : ",$Order(elves("")),!
}
```

Look here for all our solutions so far : https://bitbucket.org/bertsarens/advent2016 and https://github.com/DannyWijnschenk/AdventOfCode2016

Here is the list of all Advent of Code 2016 articles :

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

#Caché #Code Snippet #Contest #ObjectScript

Source URL:https://community.intersystems.com/post/advent-code-2016-day19-elephant-named-joseph