Article
Danny Wijnschenk · Nov 24, 2017 11m read

Advent of Code 2016 Day24: Air Duct Spelunking

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, you need to find your way through a maze (again). There are 8 points of interest in the maze, and you have to visit them all, starting with point 0.
You may visit some points more than once, in random order. The challenge is to find the shortest path through the maze while visiting all points.

The maze looks a bit like this :

```
#####################################################################################
#####################################################################################
###########
#...........#...#.....#.....#.....#....7
#.................#.............#.....#.#.#...#.#.#.......#.......#.........#.....#..
...#...#.....#.#.#.....#.......#.#.....#.....#.#.......#
#.#####.###.#####.###.#.#.#.#.#.#.#.###.###.#.#.#######.###.#.#####.#.#.#.#.###.#.#.#
.#.#.#.#####.#.#.###.#.###.#.#.#.###.#.#.###.#.#.#####.#.#.#.#.#####.###.#.#.###.#.#.
#.#.#.###.#
#.......#...#...#.#.....#.#.#...#...#...#.....#...#.......#.......#.......#.#.....#.#
.....#.........#...#...#.#...#.#.#.....#.....#.......#.#...#.#.......#...#.#...#.#...
#.#.#.....#
#.#.#.###.#.#.#.###.###.#.#.###.#.###.#.#.#.###.#.#.#.#.#.#######.#.#####.#.#.###.#.#
.###.#.#.#####.#.#.#####.#.#.#.#.#.#.#.#.#.#####.#.#.###.#.#.#.###.###.#.#.#.#.#.#.#.
#.#.#.#.#.#
#...........#.#.....#.....#...#...#.#.....#.#.#.........#.....#...#...#...#...#...#.#
.....#.#.....#...#.........#...#.........#.#.#...............#.....#.#...#.#...#.#...
....#.#...#
#.#####.###.#.###########.#.#.#.#.#.#.#.###.#.#######.#.#.#.#.#.###.#.#.#.#.#.#.#.###
.#.#.#.#########.###.#.#####.#.#############.#.#.#.#.#.#.###.#.#.#.#.###.#.#.#.#.#.##
###.#.###.#
#...#.#.....#....4
#.......#...#.......#.......#...#.......#...#.#...#...#...#.....#.#.#...#.....#...#..
.#...#...#...#.............#...#...#...#...#.#...#.#...#...#.#...#..0#.....#.#
#.#.#.###.#.#.#.#.###.###.#.###.#####.#.###.#.#.#.#.#####.#.#.###.###.#.###.###.###.#
##.#.#.#.#.###.#.#.#.#.#.#####.#.#.#####.#.#.###.#.#.#.###.#.#.#.#.###.#.#.#####.#.#.
###.#.#.###
#.#...#...........#...#.....#.#...#.#.#.#.#.......#...........#.....#.....#.#.#.#....
...#.#...#.....#.#.......#.....#...#.......#.#.....#.#.#.....#.#.#.#.#.#.#.........#.
#...#.#.#.#
#.#.#.#.#########.###.#.#.###.#.###.#####.###.#.#.#.###.#.#.###.###.###.#.###.#######
##.#.###.#.#.#.#.#.#.#.#.#########.#.#####.#.#####.#.#.###.#.#.#.#.#.#####.###.#.#.#.
###.#.#.#.#
#.#.....#.....#.......#.#...#.....#...........#.#.#.....#...#.#.........#.#.#.#......
...#.#.......#.......#...#...#...#...#.#...........#.#.....#.#.......#...........#...
#...#.....#
```

```
#.#####.#.#.#.#####.#.#.#.#.#.#.#.#.#.#.#.###.###.###.#####.###.###.#.###.#.#.#.###
.#######.###.#.#.#.#####.#.#####.#.###.#.#####.#.#.#.#########.#.###.#.#.#.#.#.#.###.
#####.###.#
#.#...#.........#.....#...#.......#...#.#...#.......#...#...#.#.......#.......#...#..
...#...........#...#.#.....#.................#.#...#.#.........#.#...#.#.#.#.......#.
#.....#...#
#.#.#.#.#.###.#.#.###.#.#.#.#####.#.###.#.###.###.###.#######.#.###.###.#.###.#.#.###
.#.#.#####.###.#.#.#.#.###.#.###.#######.#.#.#.#.#.#.#.#.#.#######.#.#.#.#.###.#.#.#.
###.#.#.###
#.....#.#...#.....#...#.#.#.....#.#.#...#.#.#...........#.....#.#...#..............
.#.#.#...#.........#.......#.#...#.......#.#.#.....#...#...........#.#...#.........#.
..........#
###.#.#.#.#.###.#.#.#.#.#.#.#.#.#.#.#.#.#.#.#.#.#.#####.###.###.#.###.#.#.#.###.#.#####
##.#.#####.###.#.#.#.#.#.#.#####.#.#.#.###.#.#.#######.#.#.#.#####.#.###.#.###.###.#.
#.###.#####
#.#.#.#.#...#.....#...#...#...#...#.#...#.#.#.......#...#.#..............#...#.....#
.....#.....#.....#.#.#.......#...#.#.....#.....#.........#...#...#.#.......#...#.....
#.....#...#
#.#.#.#.###.###.#.#.#.#########.#.#.#########.#.#.###.#.#.#.###.#####.###.#.#.#.#.###
.#.#.#.#.###.#.#.#.#.#.#.#####.###.#.#.#.###.#.#.#.#.#.#.###.#.#.#.###.#.###.#####.##
#.#.#.#.#.#
#.....#.....#.....#.....#.......#...........#...#...#.#.....#.#.......#.......#.#.#.......
.#...
..........
#...#.#.#.....#.....#...#...#.......#...#.......#...#...#...#....1#.....#...#.#.#
###.#.#.#######.#.#.#.###.#.#.###.###.#.###.#.#.#.#.#.#.###.###.###.#.#.#.#.#.#.#.#.#
####.#####.###.#.###.#.#.#.#.###.###.#.#####.###.#.#.###.###.#######.#.###.###.#.###.
#.#.#.###.#
#...#...#5
......#.#...#.......#...#.........#.....#...........#...#.#.......#.#...........#...#
.......#.....#.#.......#...#.....#.......#...#.#................#.....#...#...#.....
#
#.###.#.#.#.#####.#####.#####.###.#.#########.###.###.#.#.#####.#.#.###.#.###.###.#.#
.#.#.#.###.#.#######.#.#.#.#.#.###.#.#.###.#.#####.###.#.#.#.###.#.#.###.#.###.######
###.#####.#
#.#.#...#...#.....#.....#.#.....#.#.....#.........#...#...#.....#.#.#.......#.......#
.......#...#.........#...#...#.....#.#.#...#.........#.#.#.#.....#.#.....#.........
#.....#...#
###.###.###.#.#.#.#.###.#.#.#.#.#.#.#.#.#.#####.#.#.#.#.###.#.#####.#.###.#.#.###.#.#
.#.#.#.###.#.#.###.#.#########.#.#####.###.#.###.#####.#.#.#.#.#.#.###.#.#.#.#.#.###.
#.#.#.#.#.#
#.#.#.#.......#...#.#...#.....#.#.................#.#.........#.....#...#...#.#.....#
.....#.....#...#...#.....#...#...#...#.#.#...#...#...#...#...#...#.#...#.............
..........#
#.###.##############.#.#######.#.###.############.#.#.###.###.#.#.###.###.#.###.###.#.#
.###.#.#.###.#.#.#.###.#.###.#.#.#.#.#.#.#.###.#.#.#.#.#.#.#.#.#.#.###.###.#.#.#.#.
#.#.#.#.#.#
#.....#.#.......#.............#.....#..........#...#.#.#...#...#...#.#.#...#......
.#...#...#.....#.....#.#.#.............#.#...#.....#.....#.#.#.#.......#...#.#...#.
..#.....#.#
#.#.###.#.###.#.#.#######.#.###.#.###.#.#.#.#.#####.#.#.#.#.#.#.#.###.#######.#.#.###
.###.#.#.#.#######.#.###.#.#.############.###.#########.#.#####.#.#.#.#.#.#.#.#.#.####
#.###.#.###
#...#...#.#.............#.#.......#.#...#.#.......#.#...#.#.....#.....#.#.......#...
.......#.#.#...#.#.................#...#...#...........#.......#.......#.#...#...
#.....#.#.#
#.#.#####.#.#######.#.###.#.#####.#.#.#.#.###.#.#.#.###.#.#.#####.###.###.#.#.#.#.#.#
####.#.###.#.#.#.#.###.#.#####.#######.#.###.#.#.#.#.#####.#.#####.###.#######.#.#.#.
#.###.#.#.#
```

```
#.........#.#...#.#.......#...#...#..........#.....#.....#.......#.....#.....#.#.#..
.#.#...#.....#.....#.....#...#.#.........#.....#.....#.#.#.....#...#...#.......#.#.#.
....#...#.#
###.#######.#.#.#.#.###.#.###.#.###.###.#.###.#.#.#.#.#.#.###.#.###.#.#.#.###.###.#.#
.#.#.###.#.#.###.#.#.#.#.#.#.#.#.#.#####.#####.###.#.#.#.#####.###.###.#.#.#####.#.
#####.#.###
#...#.......#.......#...#...#.........#.....#...#...#...#.#...#.........#...#.....#..
.#.#.
#.#.#...#.
....#.#.....#.....#.#.........#.....#.#.#.....#...#...#.......#.#3....#...#.....#
#.###.###.#####.###.###.###.#.###.#.###.###.#.###.###.#.#.#######.#.#.#.#.###.#.#.#.#
##.###.#.#.###.#.#.#.#####.###.###.###.#.###.#.#.###.#.#.#####.###.#.#.###.#.#.###.#.
#.#.###.#.#
#.#...#.....#.#...#...#.#...........#...#.#...#.#.#.....#.#.......#.#.....#...#...#..
...#...#.......#.........#...#.#...#.#.#.#.....#.....#.......#.........#.....#.....#.
#...#.#...#
#.#.#.#####.#.############.###.#.#####.###.###.#.###.###.#.#.#####.#.#.#.#.#.#.#.#.#
.#.#.#.#.#.#.#.#.#.###.#.###.#.#.#.#.###.#.#.#.#######################.###.###.###.
#.#.#.#.#.#
#.#...#.#...#.....#...#...#.....#.......#.......#.....#.#...#.....#...#.....#...#....
...#...#.#...#.#...#.#...........#...#.#...#.#.......#.........#.#.......#..........
#...#...#.#
###.###.#.#.#.#.#.#.#####.#.#.#.###.#.#######.#.###.#.#####.#.###.###.#.#.###.#.#.#.#
##.#.#######.#.#.#.###.###.#.#######.#.#.#.#.#.#.###.#.#.#.#.#.#.#.#.#.#.#.###.#.#.###.
###.#.#.###
#.#...#.......#.#...#.#.........#.........#...#.#...#.........#.#...#...#.........#..
.#.....#.......#...#.....#.....#.......
....#..2#.....#.#.....#...#.#.....#.......#...#...#...#.#
#.#.#.#.#.#.#.#.###.#.###.#.###.#.#.###.#.#.#.#.#.#.###.#.#####.#.###.###.#.###.#.#.#
.#####.#.#.###.#.###.###.###.#.#####.#.#####.#####.#.#.#.#.#.###.#.#.#####.#.#.#.#.##
#.#.#.###.#
#.#..6
#.........#...#.#.......#.#.......#.........#...#.....#.#...#...#.#.#.....#.....#.#..
.....................#.......#...#.#.........#.#...#..........#...#.....#.....#...
#...#
###############################################################################################
###############################################################################################
###########
```

I decided to break down this challenge in two parts :

- find the shortest path between all pairs of points of interest (0-1, 0-2, 0-3, ..., 1-2, ..., 6-7)

You can find this code in method Points

- calculate all possible paths and take the shortest length (01234567, 01234576, ...)

You can find this code in method Combinations

In previous days, we already had to code our way through a maze, and we also had to calculate all possible permutations between 8 points (8! possibilities).

Here can you find the full explanation of the challenge : http://adventofcode.com/2016/day/24, which also links to your puzzle input (a very big maze with the 8 poi's).

Here is the code i wrote for this part :

```
Class AOC2016.Day24 Extends AOC2016.Utils
{

ClassMethod Part1(file As %String = "day24.txt")
{
  #Dim input as %String
  #Dim iInput, maxX, maxY, x, y, xStart, yStart, point as %Integer
  Try {
    Do ..Input(file, .input)
    //create grid, find all poi's
    Set xStart=0, yStart=0
    For y=1:1:input {
      For x=1:1:$Length(input(y)) {
        Set point=$E(input(y),x)
        Set %grid(x,y)=point
        If point="0" {  //starting point
          Set xStart=x, yStart=y
        }
        If point?1N {   //point of interest
          set point(point)=x_","_y
        }
      }
    }
    Do ..Points(.point)
    set %minLength=""
    Do ..Combinations("")
    Write "min length = ",%minLength,! //length of smallest path
    Write "minCombi = ",%minCombi,! //order of poi's in path
    Write "len combi = ",%lenCombi,! //length of all intermediate paths between poi's
    Quit
  } catch {
    Write "Error : ",$ZError,!
  }
}

/// Calculate pathlength of all pairs of points 0-7
ClassMethod Points(point)
{
  #Dim startLength as %Integer = 50 //part2 needs longer length = 200
  #Dim point2 as %Integer
  #Dim path as %String
  Set point="" For {
    Set point=$Order(point(point)) If point="" Quit
      set point2=point For {
        Set point2=$Order(point(point2)) If point2="" Quit
        If $Data(^points(point,point2)) Continue
        Set path="",%pathLength=startLength,%count=0
        write point,"->",point2,!
        Do ..Move($Piece(point(point),",",1),$Piece(point(point),",",2),$Piece(
point(point2),",",1),$Piece(point(point2),",",2),path)
        If %pathLength'=startLength {  //store the distance between points
        set ^points(point,point2)=%pathLength
        set ^points(point2,point)=%pathLength
      }
    }
  }
```

```
}

//Try all combinations of poi's, and find the one with the smallest pathlength
ClassMethod Combinations(s) As %String
{
  #Dim ok as %Boolean = 0
  #Dim lenCombi, s2 as %String
  #Dim iLetter, length, iS, len as %Integer
  #Dim letter as %Char

  If $Length(s)=7 {
    Set length=0
    Set lenCombi=""
    set s2="0"_s      //this is part1 : we need to start on poi 0
    //set s2="0"_s_"0" //this is part2 : we need to start and finish on poi 0
    For iS=2:1:$Length(s2) {
      If '$Data(^points($E(s2,iS-1),$E(s2,iS))) set length=9999999 Quit
      Set len=^points($E(s2,iS-1),$E(s2,iS))
      Set length=length+len
      Set lenCombi=lenCombi_" "_len
    }
    If (length<%minLength)!(%minLength="") Set %minLength=length,%minCombi=s,
%lenCombi=lenCombi
  } else {  //try all combinations by adding at each position one of the 8 poi's
    For iLetter=1:1:7 {
      Set letter=$E("1234567",iLetter)
      If s'[letter {
        Set ok = ..Combinations(s_letter)
        If ok Quit
      }
    }
  }
  Quit ok
}

ClassMethod Move(x, y, x1, y1, path)
{
  #Dim move as %String
  #Dim deltaX, deltaY, dist as %Integer
  #Dim sort as Array of %String
  set %count=%count+1
  If (x=x1)&(y=y1) {  //finished
    If ($ListLength(path)<%pathLength)!(%pathLength="") Set %pathLength=
$ListLength(path) Write %pathLength," "
  } elseIf ($ListLength(path)'<%pathLength)&(%pathLength'="") {
    //path is already longer than the current smallest path
    Quit
  } elseIf (($ListLength(path)+..Distance(x,y,x1,y1))'<%pathLength)&(
%pathLength'="") {
    //path + minimal distance of current location to goal is already longer than curr
ent smallest path
    Quit
  } else {
    For move="1,0","0,1","-1,0","0,-1" {
      Set deltaX=$Piece(move,",",1)
      Set deltaY=$Piece(move,",",2)
      If '$Data(%grid(x+deltaX,y+deltaY)) Continue  //already been there
      If %grid(x+deltaX,y+deltaY)="#" Continue      //we hit a wall !
      If $ListFind(path,$Lb(x+deltaX,y+deltaY)) Continue  //already in our path
```

```
      set sort(..Distance(x+deltaX,y+deltaY,x1,y1),move)=""
    }
    //take next move according to the direction that will be closest to goal
    Set dist="" For {
      Set dist=$Order(sort(dist)) if dist="" Quit
      set move="" For {
        set move=$order(sort(dist,move)) If move="" Quit
        Set deltaX=$Piece(move,",",1)
        Set deltaY=$Piece(move,",",2)
        Do ..Move(x+deltaX,y+deltaY, x1, y1, path_$Lb($Lb(x+deltaX,y+deltaY)))
      }
    }
  }
}

//calculate the 'manhattan' distance between two coordinates
ClassMethod Distance(x0, y0, x1, y1)
{
  Quit $ZAbs(x1-x0)+$ZAbs(y1-y0)
}

}
```

The second part of the challenge is a small addition : instead of ending in some random point of interest, you have to end at point 0, your starting point.
I did not have to change the code a lot, just had to add 0 at the end of every possible path.

But this had a huge impact on the performance, eventually, i got the answer, but it had cost a lot of cpu power.
Oh well, if i have some spare time, i will have to rethink the algorithm, but for now, i am glad i made it through the 24th day!

You are welcome to post a faster algorithm...

At last, tomorrow, it will be the end of this advent-of-code-2016 challenge !

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