Exercise 20: Walking the Grid

Andreas Loibl

November 10, 2010

Inhaltsverzeichnis

Thoughts

- the path through the grid can be expressed as a sequence of “right” and “down” directions

- the path through the grid can be expressed as a sequence of “right” and “down” directions
- the grid is quadratic so there will be the same amount of “right” directions as “down” directions

- the path through the grid can be expressed as a sequence of “right” and “down” directions
- the grid is quadratic so there will be the same amount of “right” directions as “down” directions
- the total count of directions is always $2 \cdot sidelength$ (one sidelength “right” and one sidelength “down”)

- the path through the grid can be expressed as a sequence of "right" and "down" directions
- the grid is quadratic so there will be the same amount of "right" directions as "down" directions
- the total count of directions is always $2 \cdot sidelength$ (one sidelength "right" and one sidelength "down")
- so all possible paths can be expressed as the permutation of something like $\{right, right, right, right, down, down, down, down\}$ (for a 4x4 grid in this case)

- the path through the grid can be expressed as a sequence of "right" and "down" directions
- the grid is quadratic so there will be the same amount of "right" directions as "down" directions
- the total count of directions is always $2 \cdot sidelength$ (one sidelength "right" and one sidelength "down")
- so all possible paths can be expressed as the permutation of something like $\{right, right, right, right, down, down, down, down\}$ (for a 4x4 grid in this case)
- using the combinatorics formula for permutations you get the total number of possible paths as the following:

$$\frac{(2n)!}{n! \cdot n!} \qquad (n : \texttt{side length})$$

- to put this into a algorithm (i.e. do not simply use R's "factorial()" function) this equation can be written as a series representation:

$$a_n = \frac{(2n)!}{(n!)^2}$$

- to put this into a algorithm (i.e. do not simply use R's "factorial()" function) this equation can be written as a series representation:

$$a_n = \frac{(2n)!}{(n!)^2}$$

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

- to put this into a algorithm (i.e. do not simply use R's "factorial()" function) this equation can be written as a series representation:

$$a_n = \frac{(2n)!}{(n!)^2}$$

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

$$\frac{a_n}{a_{n-1}} = \frac{(2n)!}{(2(n-1))!} \cdot \frac{((n-1)!)^2}{(n!)^2}$$

- to put this into a algorithm (i.e. do not simply use R's "factorial()" function) this equation can be written as a series representation:

$$a_n = \frac{(2n)!}{(n!)^2}$$

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

$$\frac{a_n}{a_{n-1}} = \frac{(2n)!}{(2(n-1))!} \cdot \frac{((n-1)!)^2}{(n!)^2}$$

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

- to put this into a algorithm (i.e. do not simply use R's "factorial()" function) this equation can be written as a series representation:

$$a_n = \frac{(2n)!}{(n!)^2}$$

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

$$\frac{a_n}{a_{n-1}} = \frac{(2n)!}{(2(n-1))!} \cdot \frac{((n-1)!)^2}{(n!)^2}$$

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

- to put this into a algorithm (i.e. do not simply use R's "factorial()" function) this equation can be written as a series representation:

$$a_n = \frac{(2n)!}{(n!)^2}$$

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

$$\frac{a_n}{a_{n-1}} = \frac{(2n)!}{(2(n-1))!} \cdot \frac{((n-1)!)^2}{(n!)^2}$$

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

$$a_n = a_{n-1} \cdot \frac{2n(2n-1)}{n^2}$$

Algorithm

in Pseudocode

```
function NumberOfGridRoutes(n)
    if n < 2 then                                              ▷ termination condition
        return 2
    end if
    return NumberOfGridRoutes(n − 1) · 2n(2n−1)/n^2            ▷ recursion
end function
```

in Pseudocode

```
function NUMBEROFGRIDROUTES(n)
    if n < 2 then                                          ▷ termination condition
        return 2
    end if
    return NUMBEROFGRIDROUTES(n − 1) · 2n(2n−1)/n^2          ▷ recursion
end function
```

in Pseudocode

```
function NumberOfGridRoutes(n)
    if n < 2 then                                   ▷ termination condition
        return 2
    end if
    return NumberOfGridRoutes(n − 1) · 2n(2n−1)/n^2 ▷ recursion
end function
```

in R code

```
number_of_grid_routes <- function(n)
{
  if(n<2) return(2)
  return(number_of_grid_routes(n-1) * (2*n * (2*n
      -1)) / (n * n));
}
```

Question

How many routes are there for a side of length 6, 12 or 18?

```
> source("ex20_walking_the_grid.R")
> number_of_grid_routes(6)
[1] 924
> number_of_grid_routes(12)
[1] 2704156
> number_of_grid_routes(18)
[1] 9075135300
```

Answer

There are **924**, **2,704,156** and **9,075,135,300** routes for a side length of 6, 12 and 18.