Exercise sheet n°1 : Basic concepts

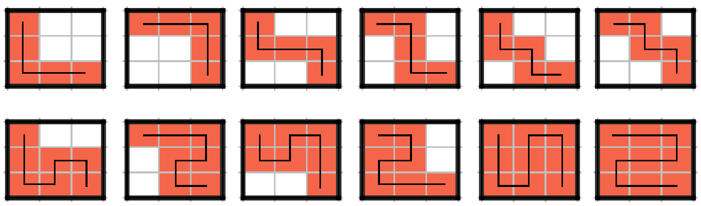
1.1 Theoretical part

Exercice 1 :

Let $G^{3\times 3}$ be a finite grid.

(a) How many different 4-connected paths from grid cell (0,0) to grid cell (2,2) exist?

Solution: 12:



(b) How many different 8-connected paths from grid cell (0,0) to grid cell (2,2) exist?

Solution: Already almost impossible to count by hand: 235.

(c) Can you answer the two questions above for a 4×4 grid and a 5×5 grid?

Solution: There exists no closed formula for this. The sequences are nevertheless known (paths on $n \times n$ grid for $n = 1, \dots, 5$):
4-connected: `https://oeis.org/A007764` - 1, 2, 12, 184, 8512
8-connected: `https://oeis.org/A140518` - 1, 5, 235, 96371, 447544629
Take-home-message: "check all possible paths" is no promising strategy for anything.

(d) Let $M = \{1, 2, 3, 4, 5, 6, 7, 8, 9, \perp\}$. How many different datasets can be definend that map G to M?

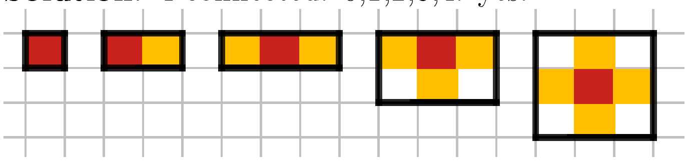
Solution: For the each grid cell there are 10 possibile values. As there are 9 grid cells the result is 10^9

Exercice 2 :

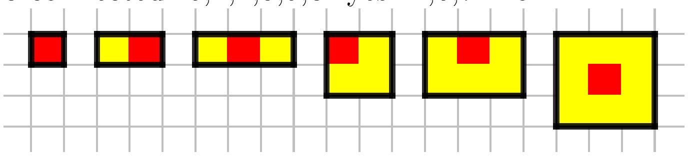
In the lecture we saw two inequalities for the cardinality of neighbourhoods: $\forall G, i, j : 0 \leq |M^4_G(i,j)| \leq 4$ and $\forall G, i, j : 0 \leq |M^8_G(i,j)| \leq 8$. Answer (and argue) for each integer $k = 0, \dots, 4$ resp. $k = 0, \dots, 8$:

(a) Is there a triple G, i, j such that $|M^4_G(i,j)| = k$ resp. $|M^8_G(i,j)| = k$?

Solution: 4-connected: 0,1,2,3,4: yes.



8-connected: 0,1,2,3,5,8: yes. 4,6,7: no



Exercice 3 :

In the lecture we saw how neighbourhoods can be seen as relations. As an example lets consider M^8_G and define a relation $\mathcal{R}$ as $((i_1, j_1), (i_2, j_2)) \in \mathcal{R} \iff (i_1, j_1) \in M^8_G(i_2, j_2)$. Answer and argue:

(a) Is $\mathcal{R}$ reflexive (that is $\forall i : (i, i) \in \mathcal{R}$)?

Solution: No, by definition.

(b) Is $\mathcal{R}$ irreflexive (that is $\forall i : (i, i) \notin \mathcal{R}$)?

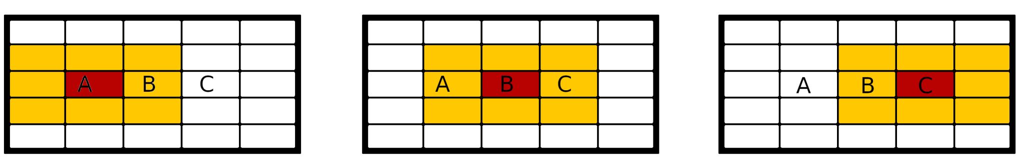
Solution: Yes, by definition.

(c) Is $\mathcal{R}$ symmetrical (that is $\forall i, j : (i, j) \in \mathcal{R} \rightarrow (j, i) \in \mathcal{R}$)?

Solution: Yes. $(i_1, j_1) \in M^8_G(i_2, j_2)$ means by definition that one of eight cases occurs. Case one: $i_1 = i_2 - 1$ and $j_1 = j_2$ Then: $i_2 = i_1 + 1$ and $j_2 = j_1$. And thus: $(i_2, j_2) \in M^8_G(i_1, j_1)$. The other cases can be threated similar (has not to be done in detail).

(d) Is $\mathcal{R}$ transitive (that is $\forall i, j, k : (i, j) \in \mathcal{R}$ and $(j, k) \in \mathcal{R} \rightarrow (i, k) \in \mathcal{R}$)?

Solution: No, by example:



B is in the 8-neighbourhood of A and C is in the 8-neighbourhood of B but C is NOT in the 8-neighbourhood of A.

Exercice 4 :

Connectivity can also be seen as a relation. Lets consider 8-connectivity on a dataset *data* and define $\mathcal{S}$ as $((i_1, j_1), (i_2, j_2)) \in \mathcal{S} \iff (i_1, j_1)$ is 8-connected to (i_2, j_2)

(a) Show that $\mathcal{S}$ is an equivalence relation (that is $\mathcal{S}$ is reflexive, symmetrical, transitive).

Solution: Reflexivity: $[(i, j)]$ is an (8-connected) path from (i, j) to (i, j). Symmetry: $((i_1, j_1), (i_2, j_2)) \in \mathcal{S}$ means there is an 8-connected path $[(i_1, j_1), gc2, \ldots, (i_2, j_2)]$ from (i_1, j_1) to (i_2, j_2). Then the reverse path $[(i_2, j_2), \ldots, gc2, (i_1, j_1)]$ is an 8-connected path from (i_2, j_2) to (i_1, j_1) and thus $((i_2, j_2), (i_1, j_1)) \in \mathcal{S}$. Transitivity: $((i_1, j_1), (i_2, j_2)) \in \mathcal{S}$ and $((i_2, j_2), (i_3, j_3)) \in \mathcal{S}$ means there are 8-connected paths $[(i_1, j_1), gc2, \ldots, (i_2, j_2)]$ from (i_1, j_1) to (i_2, j_2) and $[(i_2, j_2), gd2, \ldots, (i_3, j_3)]$ from (i_2, j_2) to (i_3, j_3). Then the concatenated path $[(i_1, j_1), gc2, \ldots, (i_2, j_2), gd2, \ldots, (i_3, j_3)]$ is an 8-connected path from (i_1, j_1) to (i_3, j_3) and thus $((i_1, j_1), (i_3, j_3)) \in \mathcal{S}$.

(b) Lets assume the underlying dataset is a DEM - what do the equivalence classes represent?

Solution: Two gridcells are in relation (and thus in the same equivalence class) if they are 8-connected. Thus equivalence classes represent islands.

Exercice 5 :

Let $G^{21\times21}$ be a finite grid and *data* the dataset on this grid that is shown in the figure below. Let further $\leq_7$ be a predicate with $\leq_7 (x) = (x \leq 7)$ and $\leq_8$ defined similarly. Answer the following questions and prove your answer (graphical proves are accepted):

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
1	⊥	0	2	1	0	5	7	8	8	3	6	2	6	8	4	5	2	5	1	6	⊥
2	⊥	6	6	5	2	8	7	7	4	3	4	9	2	6	4	1	4	3	7	4	⊥
3	⊥	7	2	2	8	1	4	5	4	2	1	3	1	5	5	7	8	7	0	6	⊥
4	⊥	2	3	9	0	5	8	5	4	7	9	0	2	4	7	9	1	0	2	6	⊥
5	⊥	9	8	9	8	3	1	9	4	2	6	4	7	4	7	3	4	8	6	3	⊥
6	⊥	6	5	3	8	1	3	7	3	7	3	3	4	2	2	9	0	2	6	2	⊥
7	⊥	3	4	2	9	8	0	9	3	8	6	9	7	2	2	4	5	6	1	4	⊥
8	⊥	2	7	8	9	4	5	1	0	3	0	2	7	9	8	3	1	4	1	4	⊥
9	⊥	8	8	2	9	5	4	4	0	8	9	3	2	8	3	3	5	2	2	6	⊥
10	⊥	7	4	4	8	4	0	1	8	9	5	6	6	8	9	8	0	8	5	9	⊥
11	⊥	3	1	0	9	8	9	8	9	8	9	9	1	9	0	9	0	9	9	7	⊥
12	⊥	9	5	9	4	1	7	9	1	4	8	9	1	2	9	8	4	1	2	5	⊥
13	⊥	6	4	2	4	7	9	2	6	4	5	8	2	5	4	9	5	0	3	4	⊥
14	⊥	1	6	7	7	3	4	8	4	3	1	8	9	9	9	8	8	9	9	8	⊥
15	⊥	9	8	0	4	5	0	3	7	0	5	0	3	8	5	3	1	0	0	9	⊥
16	⊥	0	4	1	9	9	8	5	4	6	7	4	1	2	6	5	3	3	0	4	⊥
17	⊥	5	2	8	7	0	1	9	5	6	5	2	7	6	2	7	4	8	0	7	⊥
18	⊥	5	0	9	0	2	5	5	0	6	3	7	5	6	3	5	6	0	1	0	⊥
19	⊥	2	8	7	0	9	8	9	9	7	0	0	6	4	2	8	1	9	6	7	⊥
20	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥

(a) Is there a $\leq_7$-constrained 4-connected path from grid cell $(1, 1)$ to grid cell $(19, 19)$?

(b) Is there a $\leq_7$-constrained 8-connected path from grid cell $(1, 1)$ to grid cell $(19, 19)$?

(c) Is there a $\leq_8$-constrained 4-connected path from grid cell $(1, 1)$ to grid cell $(19, 19)$?

(d) Is there a $\leq_8$-constrained 8-connected path from grid cell $(1, 1)$ to grid cell $(19, 19)$?

(e) Is there a $\leq_7$-constrained 4-connected path from grid cell $(19, 1)$ to grid cell $(1, 19)$?

(f) Is there a $\leq_7$-constrained 8-connected path from grid cell $(19, 1)$ to grid cell $(1, 19)$?

(g) Is there a $\leq_8$-constrained 4-connected path from grid cell $(19, 1)$ to grid cell $(1, 19)$?

(h) Is there a $\leq_8$-constrained 8-connected path from grid cell $(19, 1)$ to grid cell $(1, 19)$?

(i) Can you derive a general rule when two gridcells are $\leq_\theta$-constrained connected?

Solution: Note that each 4-connected path is also an 8-connected path by definition. All $\leq_7$-constrained paths: no. All $\leq_8$-constrained paths: yes. Graphical proof:

0	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
1	⊥	0	2	1	0	5	7	8	8	3	6	2	6	8	4	5	2	5	1	6	⊥
2	⊥	6	6	5	2	8	7	7	4	3	4	9	2	6	4	1	4	3	7	4	⊥
3	⊥	7	2	2	8	1	4	5	4	2	1	3	1	5	5	7	8	7	0	6	⊥
4	⊥	2	3	9	0	5	8	5	4	7	9	0	2	4	7	9	1	0	2	6	⊥
5	⊥	9	8	9	8	3	1	9	4	2	6	4	7	4	7	3	4	8	6	3	⊥
6	⊥	6	5	3	8	1	3	7	3	7	3	3	4	2	2	9	0	2	6	2	⊥
7	⊥	3	4	2	9	8	0	9	3	8	6	9	7	2	2	4	5	6	1	4	⊥
8	⊥	2	7	8	9	4	5	1	0	3	0	2	7	9	8	3	1	4	1	4	⊥
9	⊥	8	8	2	9	5	4	4	0	8	9	3	2	8	3	3	5	2	2	6	⊥
10	⊥	7	4	4	8	4	0	1	8	9	5	6	6	8	9	8	0	8	5	9	⊥
11	⊥	3	1	0	9	8	9	8	9	8	9	9	1	9	0	9	0	9	9	7	⊥
12	⊥	9	5	9	4	1	7	9	1	4	8	9	1	2	9	8	4	1	2	5	⊥
13	⊥	6	4	2	4	7	9	2	6	4	5	8	2	5	4	9	5	0	3	4	⊥
14	⊥	1	6	7	7	3	4	8	4	3	1	8	9	9	9	8	8	9	9	8	⊥
15	⊥	9	8	0	4	5	0	3	7	0	5	0	3	8	5	3	1	0	0	9	⊥
16	⊥	0	4	1	9	9	8	5	4	6	7	4	1	2	6	5	3	3	0	4	⊥
17	⊥	5	2	8	7	0	1	9	5	6	5	2	7	6	2	7	4	8	0	7	⊥
18	⊥	5	0	9	0	2	5	5	0	6	3	7	5	6	3	5	6	0	1	0	⊥
19	⊥	2	8	7	0	9	8	9	9	7	0	0	6	4	2	8	1	9	6	7	⊥
20	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

The red gridcells define a barrier with elevation 8 or above. No $\leq_7$-constrained paths can cross this barrier. The $\leq_8$-constrained paths are shown in green and can cross the barrier (at gridcells with data values 8 - these are not colored green here to emphasize the barrier). This can be used to characterize when two gridcells are $\leq_\theta$-constrained connected: if and only if there is no barrier with data values only greater than θ between them. This could be formalised.

1.2 Practical part

Exercice 1 :

The library code is available on `https://gitlab.com/globalclimateforum/diva_library`. The Julia language can be downloaded from `https://julialang.org/`. Code can be edited with usual editors, for instance Visual Studio `https://visualstudio.microsoft.com/de/` or build-in editors like vim or emacs.

(a) Check if you can see/checkout the repository. Checkout should work as follows:
`git clone -branch exercises https://gitlab.com/globalclimateforum/diva_library.git`
At the location called a sub directory `diva_library` will be created with furter subdirectories `exercises` and therein `exercise01`. There are a few files in this directory:

- `exercies1_Intro.jl` A short demo on the representation of undefined values. Can be run with `julia exercies1_Intro.jl` (called in the directory where the file is located).
- `exercies1_functions_implemented.jl` Pre-implemented functions that can be used solving the exercise. Provided are especially two neghbourhood functions `nh4` and `nh8`:

```
include("exercies1_functions_implemented.jl")
nh8 (generic function with 2 methods)

julia> A = [1 2 2 3 4; 1 2 3 3 4; 5 6 7 7 8; 9 9 8 8 8; 0 0 0 0 0]
5×5 Matrix{Int64}:
 1  2  2  3  4
 1  2  3  3  4
 5  6  7  7  8
 9  9  8  8  8
 0  0  0  0  0

julia> nh4(A,3,4)
4-element Vector{Tuple{Int64, Int64}}:
 (2, 4)
 (4, 4)
 (3, 3)
 (3, 5)

julia> nh8(A,(3,4))
8-element Vector{Tuple{Int64, Int64}}:
 (2, 4)
 (4, 4)
 (3, 3)
 (3, 5)
 (2, 3)
 (2, 5)
 (4, 3)
 (4, 5)
```

- `exercies1_functions_to_implement.jl` This file contains the functions (more accurate: the definition of the functions) you want to implement. In particular:

```
# Test if path, given as an array of pairs,
# is a valid path from startcell to endcell ind dataset
function valid_4connected_path(dataset :: Array{Float64},
    path :: Array{Tuple{Int64,Int64}},
```

```
    startcell :: Tuple{Int64,Int64},
    endcell :: Tuple{Int64,Int64}) :: Bool

end


function valid_8connected_path(dataset :: Array{Float64},
    path :: Array{Tuple{Int64,Int64}},
    startcell :: Tuple{Int64,Int64},
    endcell :: Tuple{Int64,Int64}) :: Bool

end


# Compute the coastline of a given land/elevation dataset
function coastline(dataset  :: Array{Float64}) :: Array{Bool}

end
```

- `exercies1_run.jl` The actual exercise file, implements soem test that will test if you implemented the required fucntions successfully. Can be executed the command `julia exercies1_Intro.jl` wich will produce an output like:

```
...
Test Summary:                                                                                                 | Error  Total  Time
Exercise 01                                                                                                   |     8      8  2.0s
  path1 is a valid 8-connected path from (2,2) to (6,3) but not a valid 4-connected path from (2,2) to (6,3) in data       |     2      2  1.3s
  path2 is neither a valid 8-connected path from (2,2) to (3,9) but nor a valid 4-connected path from (2,2) to (3,9) in data |     2      2  0.2s
  path3 is neither a valid 8-connected path from (2,1) to (2,8) but nor a valid 4-connected path from (2,1) to (2,8) in data |     2      2  0.2s
  data_cl should be the coastline of data                                                                     |     1      1  0.1s
  data_cl should be the coastline of random data                                                              |     1      1  0.2s
ERROR: LoadError: Some tests did not pass: 0 passed, 0 failed, 8 errored, 0 broken.
```

Your goal is that it produces the following output:

```
Test Summary:        | Pass  Total  Time
check Exercise 01    |    8      8  0.3s
```

(b) Implement the functions `valid_4connected_path` and `valid_8connected_path` such that they test if a given path is a valid path between two gridcells in a dataset. Use the definition provided in the lecture.

Solution:

```
function valid_4connected_path(dataset :: Array{Float64},
  path :: Array{Tuple{Int64,Int64}},
  startcell :: Tuple{Int64,Int64},
  endcell :: Tuple{Int64,Int64}) :: Bool

  if (path[1] != startcell) return false end
  if (path[size(path,1)] != endcell) return false end
```

```
  if dataset[path[1][1],path[1][2]]==-Inf return false end

  for i in 2:size(path,1)
    if dataset[path[i][1],path[i][2]]==-Inf return false end
    if !(path[i] in nh4(dataset,path[i-1])) return false end
  end
  return true
end

function valid_8connected_path(dataset :: Array{Float64},
  path :: Array{Tuple{Int64,Int64}},
  startcell :: Tuple{Int64,Int64},
  endcell :: Tuple{Int64,Int64}) :: Bool

  if (path[1] != startcell) return false end
  if (path[size(path,1)] != endcell) return false end
  if dataset[path[1][1],path[1][2]]==-Inf return false end

  for i in 2:size(path,1)
    if dataset[path[i][1],path[i][2]]==-Inf return false end
    if !(path[i] in nh8(dataset,path[i-1])) return false end
  end
  return true
end
```

(c) Implement the function `coastline` that computes the coastline of a given dataset (interpreted as DEM with land/ocean).

Solution:

```
function coastline(dataset  :: Array{Float64}) :: Array{Bool}
  cl  :: Array{Bool} = Array{Bool}(undef, size(dataset,1), size(dataset,2))
  for i in 1:size(dataset,1)
    for j in 1:size(dataset,2)
      if (dataset[i,j]!=-Inf)
        for nhc in nh8(dataset, i, j)
      if dataset[nhc[1],nhc[2]]==-Inf cl[i,j]=true end
        end
      end
    end
  end
  return cl
end
```