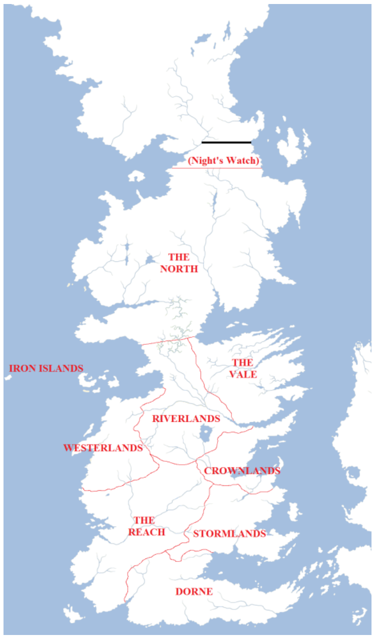
Task AI. The region coloring problem: `Westeros`

- Student name: Yori Verbist
- Number: R0697651

First find a blank map (on the internet). Choose a map you're interested in: a map of your home town, the country you were on vacation, a map used in your favorite game, the continent you would like to visit someday, ... It goes without saying that you must have enough regions (at least 10). Show the blank map below:

`insert map`



`Explain briefly what this map is all about and why you have chosen this map.`

It's the map from Game of Thrones. I chose this map because the map of Skyrim was to small.

Your goal is to color the map with the least colors possible so that no adjacent regions have the same color.

First import the classes, define the variables and the possible values that every variable can take. Choose your own colors.

In [2]:

```
from simpleai.search import CspProblem, backtrack
variables = ("Night's Watch", "The North", "The Vale",
             "Iron Islands", "Riverlands", "Westerlands",
             "Crownlands", "The Reach", "Stormlands",
             "Dorne"
            )
colors = ["red", "white", "blue", "yellow"]
domains = {
    "Night's Watch": colors,
    "The North": colors,
    "The Vale": colors,
    "Iron Islands": colors,
    "Riverlands": colors,
    "Westerlands": colors,
    "Crownlands": colors,
    "The Reach": colors,
    "Stormlands": colors,
    "Dorne": colors
}
```

Define the constraint function that imposes that two neighbors should be colored differently. Apply the constraint for every pair of regions from the chosen map.

In [3]:

```
def constraint_unique(variables,values):
    for i in range(1,len(values)):
        if(values[0] == values[i]):
            return False
    return True

constraints = [
    (("Night's Watch", "The North"), constraint_unique),
    (("The North", "Night's Watch", "The Vale", "Riverlands", "Iron Isl
    (("Iron Islands", "The North", "Riverlands", "Westerlands"), constr
    (("The Vale", "Riverlands", "Crownlands", "The North"), constraint_
    (("Riverlands", "The Vale", "The North", "Iron Islands", "Westerlar
    (("Westerlands", "Iron Islands", "Riverlands", "The Reach"), constr
    (("The Reach", "Westerlands", "Riverlands", "Crownlands", "Stormlar
    (("Crownlands", "The Vale", "Riverlands", "The Reach", "Stormlands'
    (("Stormlands", "Crownlands", "The Reach", "Dorne"), constraint_uni
    (("Dorne", "The Reach", "Stormlands"), constraint_unique),
]


problem = CspProblem(variables, domains, constraints)

output = backtrack(problem)
print('\nSolutions:', output)
```

```
Solutions: {"Night's Watch": 'red', 'The North': 'white', 'The Vale': '
blue', 'Iron Islands': 'blue', 'Riverlands': 'yellow', 'Westerlands': '
white', 'Crownlands': 'white', 'The Reach': 'blue', 'Stormlands': 'yell
ow', 'Dorne': 'white'}
```

And finaly search for a solution and print it.

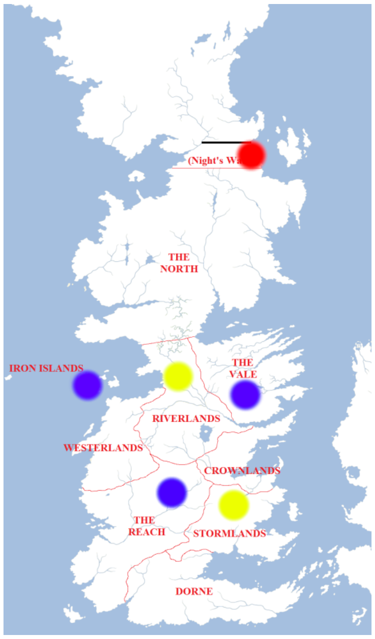
In [ ]:

```
for country, color in output.items():
    print(country, ' -> ', color )
```

Use Paint or Photoshop to color the map with the colors from the solution and check that no two adjacent regions have the same color. Show the colored map below.

`insert colored map`

Convert the resulting *jupyter notebook* to pdf. Deliver it on Canvas.


(Night's W
THE NORTH
IRON ISLANDS
THE VALE
RIVERLANDS
WESTERLANDS
CROWNLANDS
THE REACH
STORMLANDS
DORNE