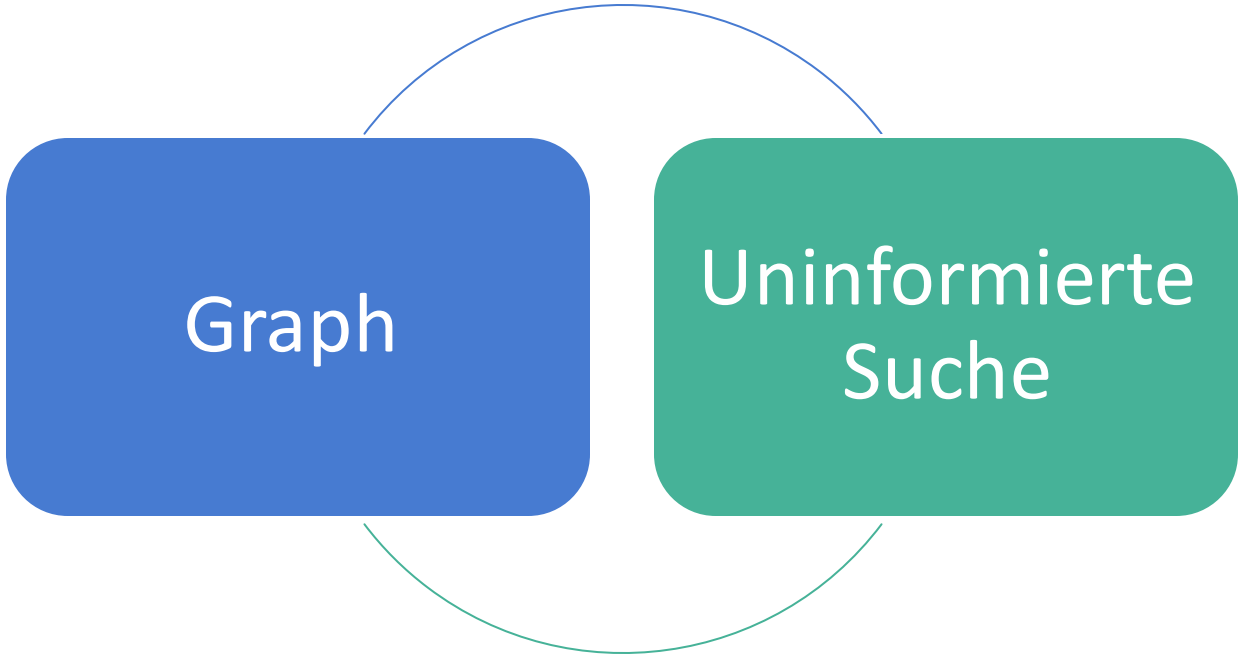
KÜNSTLICHE INTELLIGENZ ÜBUNG 5

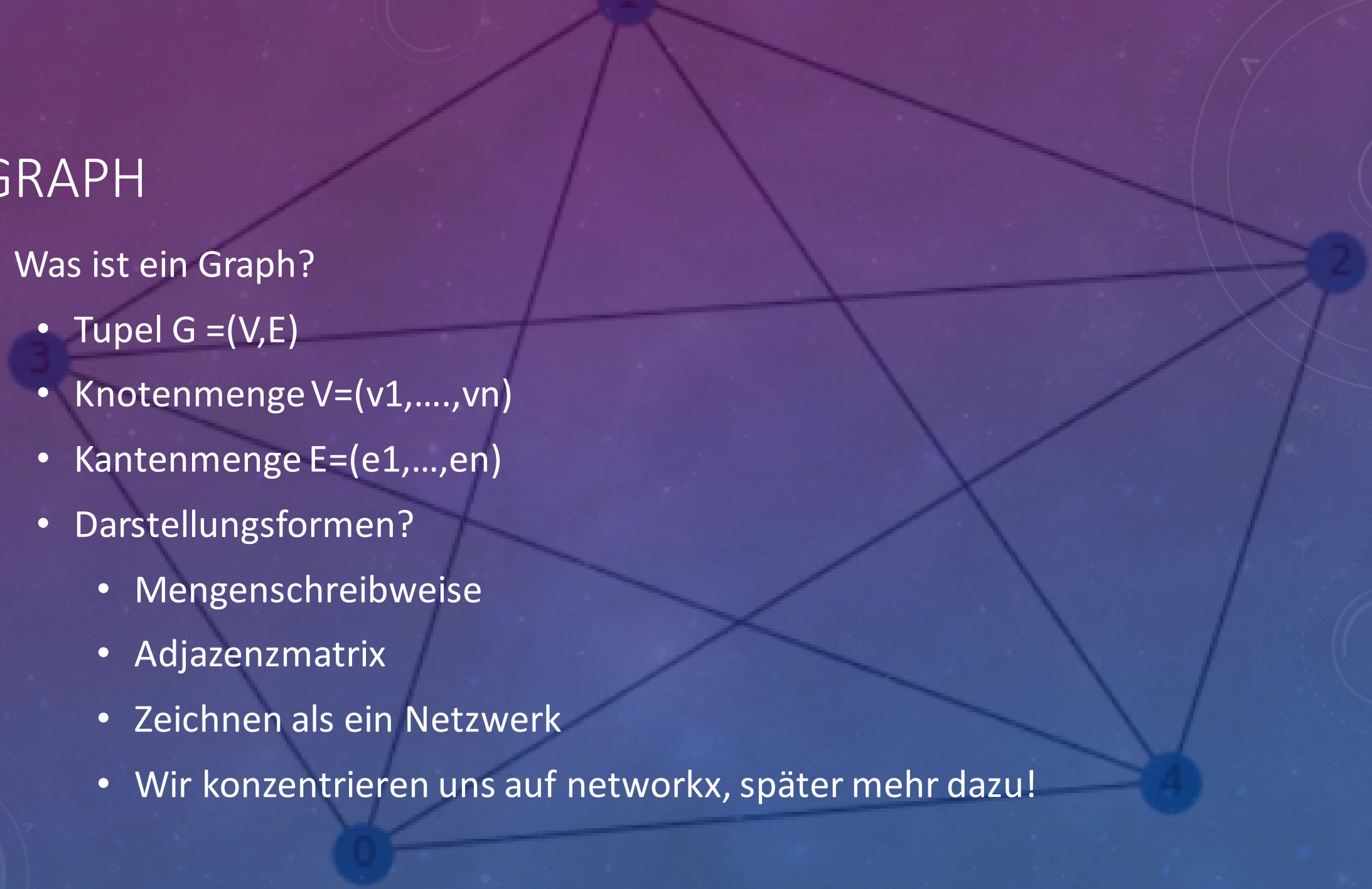
MATTHİAS TSCHÖPE, KUNAL OBEROİ

AGENDA



GRAPH

- Was ist ein Graph?
  - Tupel G =(V,E)
  - Knotenmenge V=(v1,….,vn)
  - Kantenmenge E=(e1,…,en)
  - Darstellungsformen?
    - Mengenschreibweise
    - Adjazenzmatrix
    - Zeichnen als ein Netzwerk
    - Wir konzentrieren uns auf networkx, später mehr dazu!

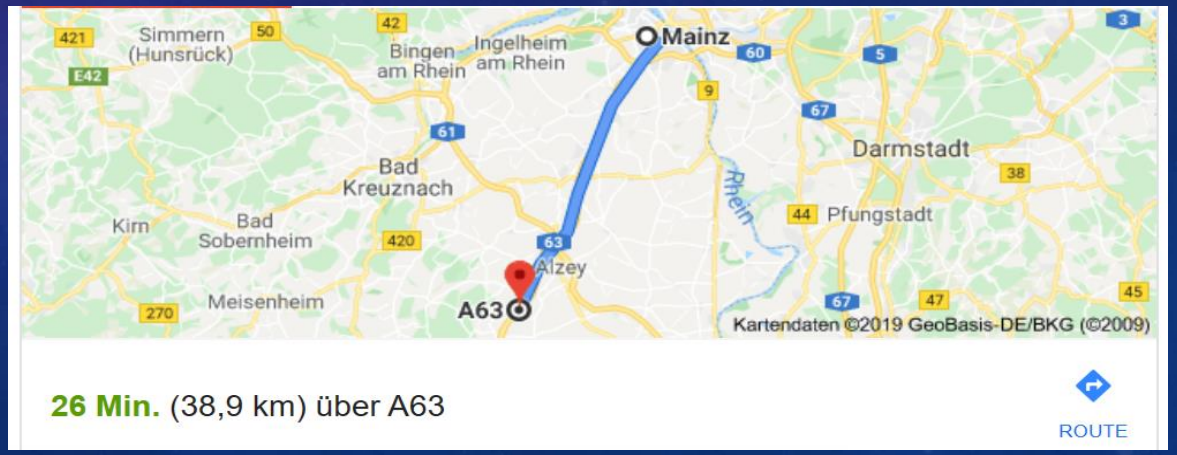


AUFGABE 1A (BASIC- GRAPH ERSTELLUNG UND PLOTTEN)

- Erstellen Sie einen Graph mit Hilfe von networkx, fügen sie mit add_node() 6 Knoten hinzu und beliebe Kanten mit add_edges()
  - Plotten sie den Graph mit draw_networkx() und übergeben Sie den Knoten die Farbe "rot"

AUFGABE 1B (STÄDTE GRAPH)

- i) Erstellen sie ein Graph und fügen sie eine Städte Liste mit 'Kaiserslautern', 'Mannheim','Mainz','Frankfurt' hinzu und Plotten Sie die Knoten
- ii) Die Verbindungen zwischen den Städten sind folgendermaßen definiert:
  - Kaiserslautern nach Mainz "81KM" (A63)
  - Kaiserslautern nach Mannheim "67KM" (A6)
  - Mainz nach Frankfurt "39KM" (A63)
  - Mannheim nach Frankfurt "80KM", über A5 und A67
  - Mainz nach Mannheim "83KM", über A61 und A63
- iii) Führen sie die Methoden dijstra_path_length() und dijstra_path() aus
  - Von 'Kaiserslautern' nach Frankfurt'
- iv) Aufgrund der Verkehrsbedingungen ist die Autobahn von Mainz nach Frankfurt gesperrt A63, welcher weg ist noch Optimal?
  - Hinweis: Entfernen sie zunächst die Verbindung zwischen Mainz und Frankfurt



UNIFORMIERTE SUCHSTRATEGIEN

- Strategien bekommen keine zusätzliche Information über Zustände (außer Informationen aus der Problemdefinition)
- Sie können einen Nachfolger und einen Zielzustand von einem Nichtzielzustand unterscheiden
- Unterscheidung der Strategien nach der Reihenfolge, in der die Knoten erweitert werden
- Beispiele: BFS und DFS es gibt noch weitere

BREADTH-FIRST-SEARCH (BFS)

- Kennzeichne jeden Knoten, Default sind alle unmarkiert
- Starte mit einem Startknoten und markiere ihn mit (Level 0)
- Nächster Schritt, finde alle Knoten die Kanten Beziehung zum Startknoten darstellen und noch nicht markiert sind (Level 1)
- Nächster Schritt ,finde alle Knoten die Kanten Beziehung zu Knoten von (Level 1) haben und markiere sie als (Level 2)
- Das machen wir solange, bis ein Level keine benachbarten Knoten mehr hat, die unmarkiert sind

DEPTH-FIRST-SEARCH (DFS)

- Kennzeichne jeden Knoten, Default sind alle unmarkiert
- Starte mit einem Startknoten und markiere ihn
- Gehe in irgendeiner Reihenfolge die zum Startknoten benachbarten Knoten durch und tue Folgendes:
  - Falls der Knoten noch nicht markiert ist, markiere ihn und starte rekursiv eine Tiefensuche von dort aus
- Zunächst wird in der Tiefe gesucht vom Startknoten aus
- DFS markiert schließlich alle Knoten, die in derselben Zusammenhangskomponenten liegen wie der Startknoten

AUFGABE 2A) (SEQUENZ SUCHE)

- a)

Führen sie die Sequenz für BFS und DFS Suche S und V.

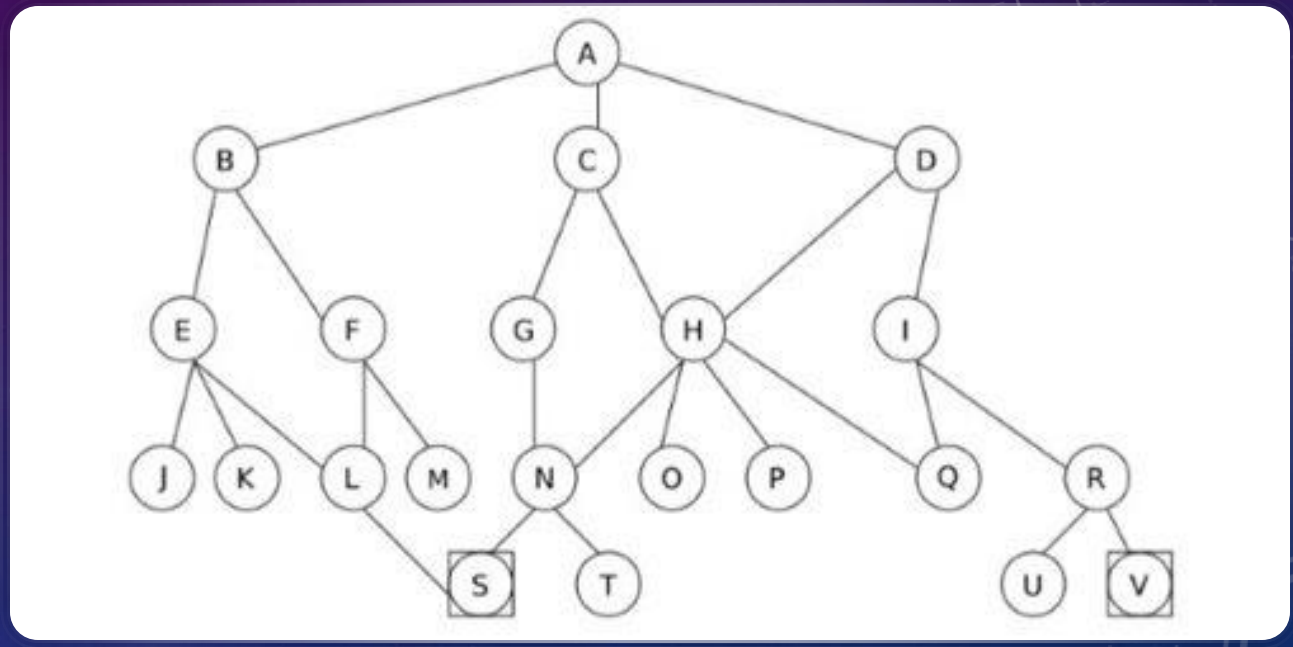
- b)

Definieren Sie "optimal" and "complete" im Kontext der Suchmethoden

Erklären sie folgende Aussagen:

BFS hat exponentielle Komplexität in der Tiefe

DFS ist nicht "optimal" und nicht "complete"

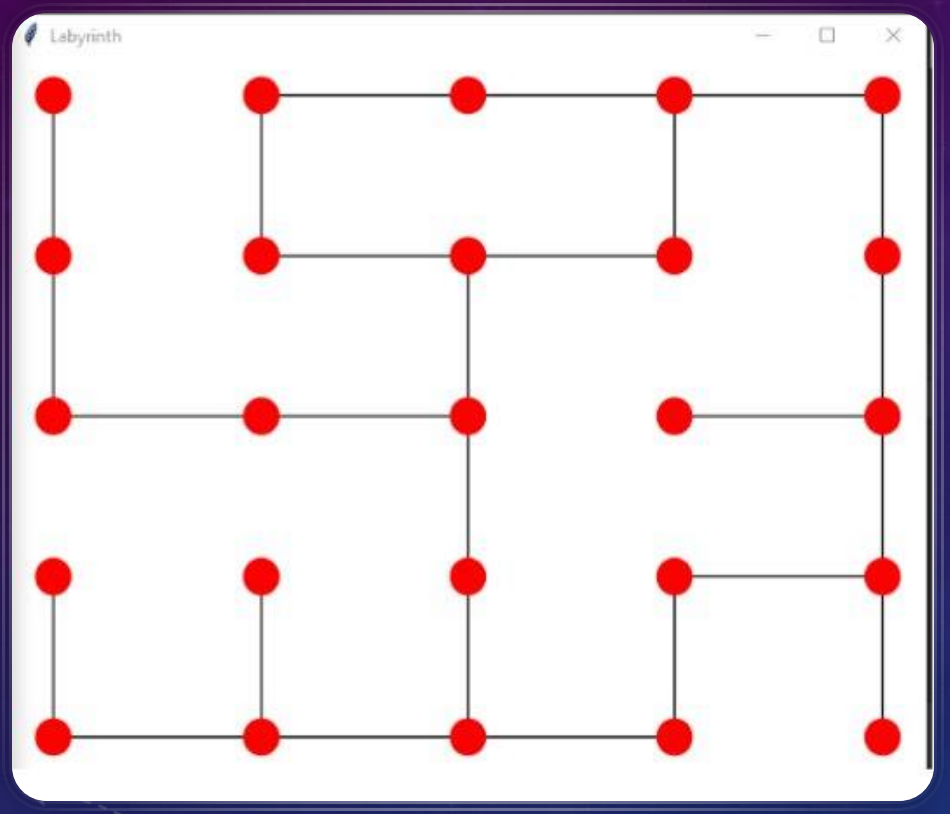


AUFGABE 2B) (BREADTH-FIRST SEARCH UND DEPTH-FIRST SERARCH)

- In der Aufgabe 2a) haben sie die Sequenz von Knoten S und V gefunden, nun um diese Theorie mehr zu vertiefen sollen Sie Breadth-First Search und Depth-First Search mit Hilfe von Python implementieren
- Bfs(graph,start) und Dfs_rec(graph,start,visited_node)
  - Übergeben sie den Graph von 2a und prüfen Sie welcher Algorithmus den Knoten 'J' schneller findet diskutieren warum das so ist.
  - Differenzieren Sie die Laufzeit beider Algorithmen
    - Der Graph kann folgendermaßen aufgebaut werden

```
graph = { 'A':['B','C','D'],
          'B':['A','E','F'],
          'C':['A','G','H'],
           .....}
```

AUFGABE 3 (LABYRINTH)



- In dieser Aufgabe wird der Weg eines Labyrinthes mit Hilfe von Graph modelliert und Lösungsansätze zum Finden eines Ausgangs entwickelt, angewendet und analysiert.
- a) Erzeugen Sie einen 5x5 GridGraph.
  - Hinweis: Der Startknoten des Labyrinthes liegt bei (4,0) und Zielknoten (0,4).
- b) Entfernen Sie Folgende Kanten um die Darstellung links zu erhalten:
- ((3,0),(4,0));((2,1),(3,1));((0,4),(1,4));((0,3),(1,3));((1,3),(1,2));((3,3),(3,2));((0,2),(0,1));((2,4),(2,3));((1,2),(1,1));((0,1),(1,1));((1,1),(2,1));((3,2),(3,1));((3,3),(4,3));((2,2),(3,2))
- c) Berechnen und Plotten Sie den DFS und BFS Baum. Was fällt Ihnen bei den Ergebnissen auf?
- d) Welchen der beiden Algorithmen würden Sie verwenden, um einen Weg durch ein Labyrinth zu finden? Begründen Sie Ihre Antwort.