

Introduction to Graphs



Warm Up: An easy question

Situation:

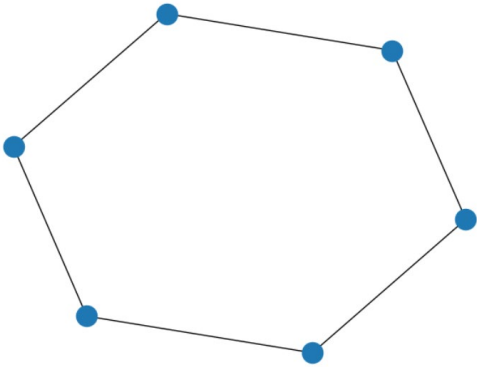
There are **n towns** that should be connected by roads. Every road is a direct connection between two towns. Every town should have a direct connection to all the other towns.

How many roads need to be build?

- a) n
- b) $n * n$
- c) $n * (n - 1)$
- d) $n * (n - 1) / 2$
- e) Depends where the towns are located.

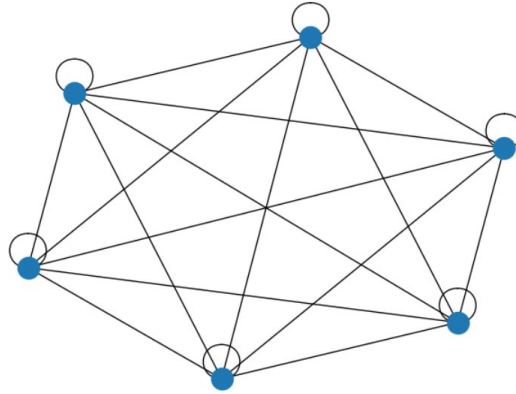
Answers

a) n



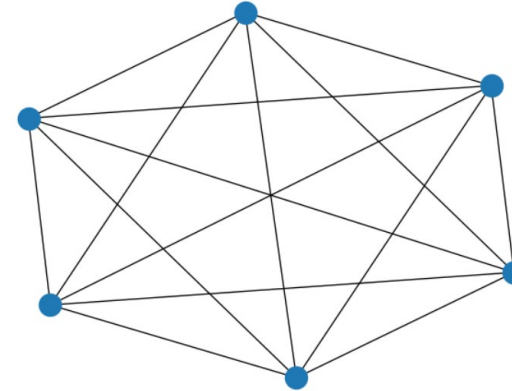
Zu wenige Straßen
Die Ortschaften sind zwar
miteinander verbunden, aber
nicht alle „direkt“.

b) $n * n$



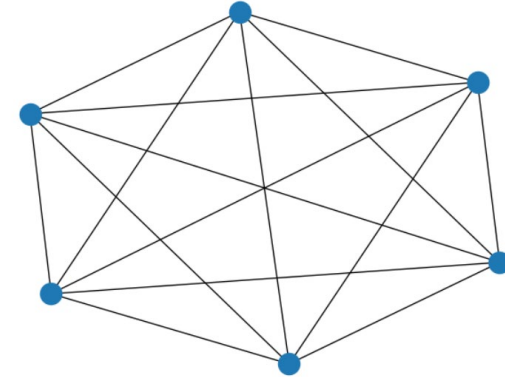
Zu viele Straßen
Eine Ortschaft muss nicht
mit sich selbst verbunden
werden.

c) $n * (n - 1)$



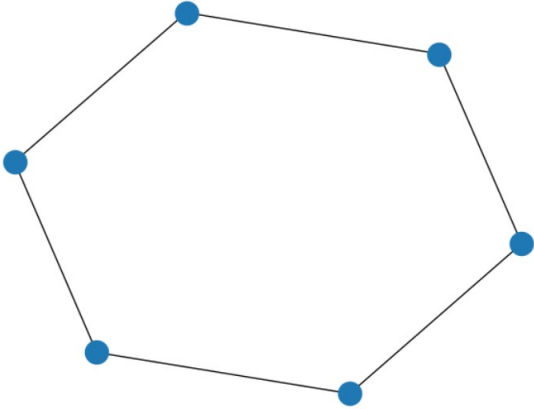
Anzahl der Straßen
erscheint richtig.
ABER: wurden ggf. Straßen
„doppelt“ gebaut?

d) $n * (n - 1) / 2$



Diese Lösung ist richtig.
Sie liefert die „**minimale**
Anzahl an Verbindungen“

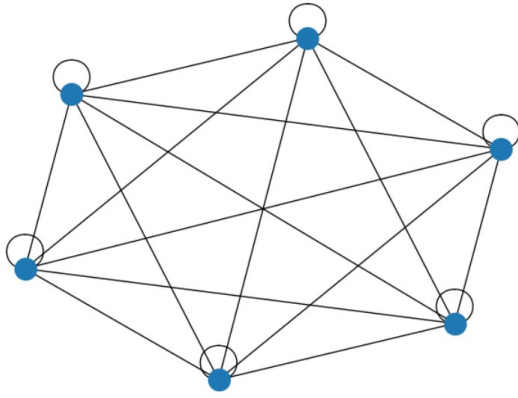
Lösung a) n



- Für (fast) alle Kanten gilt:
Die Differenz aus Nummer des Endknotens
und Nummer des Startknotens ist 1.
- Nur bei der letzten Kante hat die Differenz den
Wert -5.
- $5 * 1 + -5 = 0 \Rightarrow$ Es geht einmal im Kreis.
- Anzahl der Kanten ist identisch zu Anzahl der
Knoten

```
1 import matplotlib.pyplot as plt
2 import networkx as nx
3
4 Gnet = nx.Graph()
5 Gnet.add_nodes_from([1, 2, 3, 4, 5, 6])
6 Gnet.add_edges_from([(1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 1)])
7
8 # Draw the graph
9 nx.draw(Gnet)
10 plt.show()
11
```

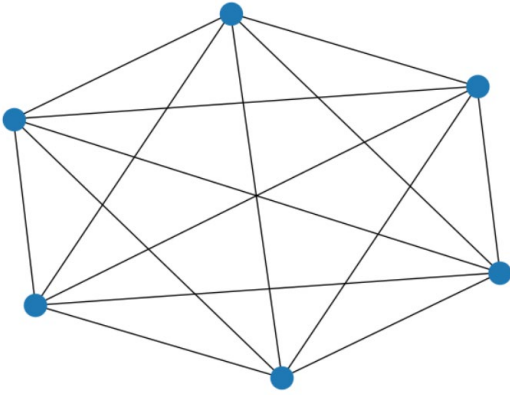
Lösung b) $n * n$



- „Simpler“ Ansatz „alle mit allen“
- Mehr geht nicht.
- Es gibt „unsinnige“ Verbindungen „auf sich selbst“, z.B. (2, 2)
- Es gibt „doppelte“ Verbindungen. Nur eine Frage der „Richtung“, z.B. (3, 4) und (4, 3)

```
1  import matplotlib.pyplot as plt
2  import networkx as nx
3
4  Gnet = nx.Graph()
5  Gnet.add_nodes_from([1, 2, 3, 4, 5, 6])
6  Gnet.add_edges_from([(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6)])
7  Gnet.add_edges_from([(2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (2, 6)])
8  Gnet.add_edges_from([(3, 1), (3, 2), (3, 3), (3, 4), (3, 5), (3, 6)])
9  Gnet.add_edges_from([(4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6)])
10 Gnet.add_edges_from([(5, 1), (5, 2), (5, 3), (5, 4), (5, 5), (5, 6)])
11 Gnet.add_edges_from([(6, 1), (6, 2), (6, 3), (6, 4), (6, 5), (6, 6)])
12
13 # Draw the graph
14 nx.draw(Gnet)
15 plt.show()
16
```

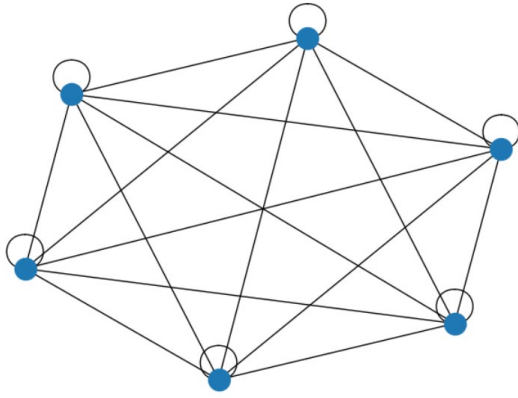
Lösung c) $n * (n - 1)$



- „Intelligenter“ Ansatz: von einer Ortschaft aus kann ich mich nur mit den anderen Orten verbinden
- Es sind $n = 6$ Ortschaften
- Damit $n - 1$ Verbindungen zu den anderen Ortschaften
- Es hat immer noch doppelte Verbindungen
- Holzweg!!!
- Nochmal zurück zu Lösung ab und genau hinsehen.

```
1  import matplotlib.pyplot as plt
2  import networkx as nx
3
4  Gnet = nx.Graph()
5  Gnet.add_nodes_from([1, 2, 3, 4, 5, 6])
6  Gnet.add_edges_from([(1, 2), (1, 3), (1, 4), (1, 5), (1, 6)])
7  Gnet.add_edges_from([(2, 1), (2, 3), (2, 4), (2, 5), (2, 6)])
8  Gnet.add_edges_from([(3, 1), (3, 2), (3, 4), (3, 5), (3, 6)])
9  Gnet.add_edges_from([(4, 1), (4, 2), (4, 3), (4, 5), (4, 6)])
10 Gnet.add_edges_from([(5, 1), (5, 2), (5, 3), (5, 4), (5, 6)])
11 Gnet.add_edges_from([(6, 1), (6, 2), (6, 3), (6, 4), (6, 5)])
12
13 # Draw the graph
14 nx.draw(Gnet)
15 plt.show()
16
```

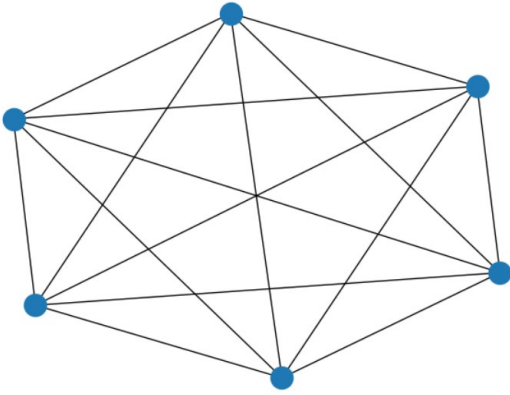
Noch einmal Lösung b) $n * n$



- Alle Verbindungen im „oberen“ Dreieck sind „überflüssig“.
- Es verbleibt das untere Dreieck mit 5 Zeilen
- Die Anzahl ergibt sich aus der Berechnung
 $1 + 2 + 3 + 4 + 5$
oder allgemein
 $1 + 2 + \dots + (n - 1)$

```
1  import matplotlib.pyplot as plt
2  import networkx as nx
3
4  Gnet = nx.Graph()
5  Gnet.add_nodes_from([1, 2, 3, 4, 5, 6])
6  Gnet.add_edges_from([(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6)])
7  Gnet.add_edges_from([(2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (2, 6)])
8  Gnet.add_edges_from([(3, 1), (3, 2), (3, 3), (3, 4), (3, 5), (3, 6)])
9  Gnet.add_edges_from([(4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6)])
10 Gnet.add_edges_from([(5, 1), (5, 2), (5, 3), (5, 4), (5, 5), (5, 6)])
11 Gnet.add_edges_from([(6, 1), (6, 2), (6, 3), (6, 4), (6, 5), (6, 6)])
12
13 # Draw the graph
14 nx.draw(Gnet)
15 plt.show()
16
```

Lösung d) Aus $1 + 2 + 3 + 4 + 5$ wird $n * (n - 1) / 2$



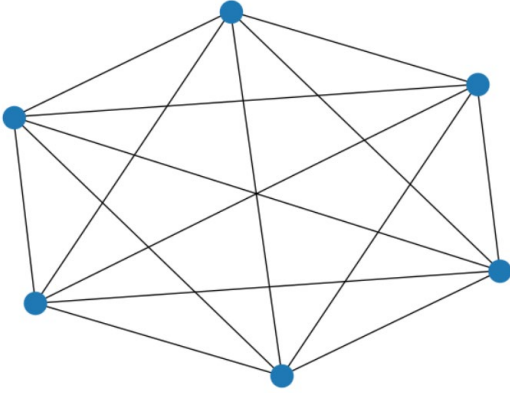
- Vorgehen:
- Von Ortschaft 1 aus baue ich Straßen zu allen anderen Ortschaften
- Von Ortschaft 2 aus baue ich Straßen zu allen anderen Ortschaften, die eine größer Nummer haben.
- Gilt das nicht für jede Ortschaft i?

```
1  import matplotlib.pyplot as plt
2  import networkx as nx
3
4  Gnet = nx.Graph()
5  Gnet.add_nodes_from([1, 2, 3, 4, 5, 6])
6  Gnet.add_edges_from([(1, 2), (1, 3), (1, 4), (1, 5), (1, 6)])
7  Gnet.add_edges_from([(2, 3), (2, 4), (2, 5), (2, 6)])
8  Gnet.add_edges_from([(3, 4), (3, 5), (3, 6)])
9  Gnet.add_edges_from([(4, 5), (4, 6)])
10 Gnet.add_edges_from([(5, 6)])
11
12 # Draw the graph
13 nx.draw(Gnet)
14 plt.show()
15
```

Hinweis: die Mathematik dahinter ist die Gauß'sche Summenformel
https://de.wikipedia.org/wiki/Gau%C3%9Fsche_Summenformel

Frage: Geht das nicht auch allgemein für jeden Wert von n ?

Antwort: ja, das geht!

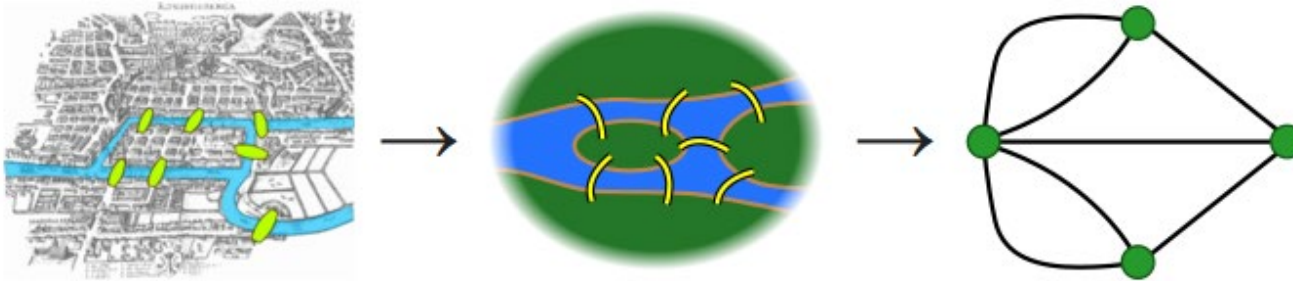


- Zeile 5: In der Variablen zählen wir mit wie viele Kanten wir erzeugt haben.
- Zeile 7: die obere Grenze des range ist nicht dabei $\Rightarrow n + 1$
- Zeile 8: wir müssen nur von $n-1$ Ortschaften aus Straßen bauen (siehe Folie zuvor)
- Zeile 9: von einem Ort i aus bauen wir nur Straßen zu Orten, welche eine größere Nummer haben.

```
1 import matplotlib.pyplot as plt
2 import networkx as nx
3
4 n = 6
5 k = 0
6 Gnet = nx.Graph()
7 Gnet.add_nodes_from(range(1, n + 1, 1))
8 for i in range(1, n, 1):
9     for j in range(i + 1, n + 1, 1):
10         k = k + 1
11         Gnet.add_edge(i, j)
12
13 print("Anzahl der Kanten: {0}".format(k) )
14 # Draw the graph
15 nx.draw(Gnet)
16 plt.show()
17
```

Ausblick: Probleme – Probleme – Probleme

Wegzoll sparen
Das Königsberger Brückenproblem



https://de.wikipedia.org/wiki/K%C3%B6nigsberger_Br%C3%BCckenproblem

Auf welchem Weg komme ich in
alle Stadtteile und muss am
wenigsten Brückenzoll zahlen?

Auf welchem Weg lege ich die
kürzeste Strecke zurück und fahre
keinen Ort mehrfach an (bis auf
Start und Ende?)

Energie sparen
Das Problem des Handlungsreisenden



https://de.wikipedia.org/wiki/Problem_des_Handlungsreisenden