# GraphGenerator

Der *GraphGenerator* erstellt einen zusammenhängenden Graphen. Knoten werden dabei zufällig miteinander verbunden.

Die Idee ist zunächst alle Knoten in eine Liste *"unconnected"* zu packen. In dieser Liste sind jetzt alle Knoten des Graphen enthalten.

In einer weiteren Liste *"connected"* sind alle Knoten, die über Kanten mit dem Graphen verbunden sind. Diese Liste ist anfangs leer.

Wir nehmen nun einen zufälligen Knoten aus der *"unconnected"* Liste und bilden eine Kante zu einem Knoten in der *"connected"* Liste. Der Knoten der *"unconnected"* List wird das dieser Liste gelöscht und in die "connected" Liste gepackt.

Wenn die Liste *"unconnected"* leer ist, dann können wir sicher sein, dass der Graph zusammenhängend ist.

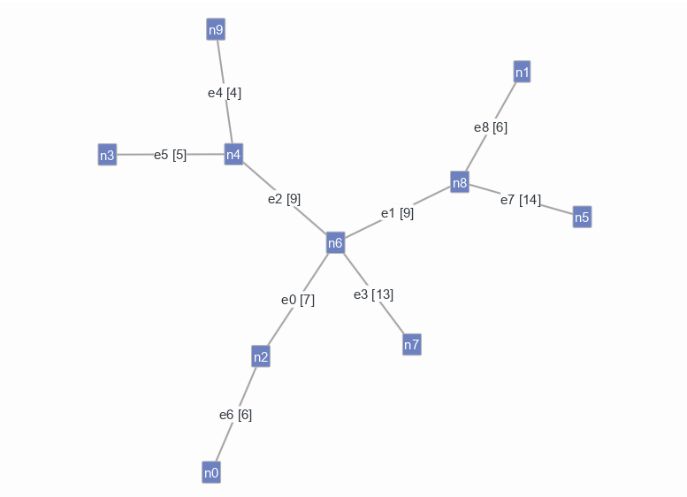


Abbildung 1 Graph nachdem Liste "unconnected" leer ist

Wenn jetzt noch Kanten erstellt werden sollen, dann nehmen wir zwei zufällig gewählte Knoten aus dem Graphen, die noch nicht verbunden sind und erstellen eine Kante. Fügt man beim oberen Graphen so weitere Kanten hinzu, dann könnte folgender Graph dabei entstehen.

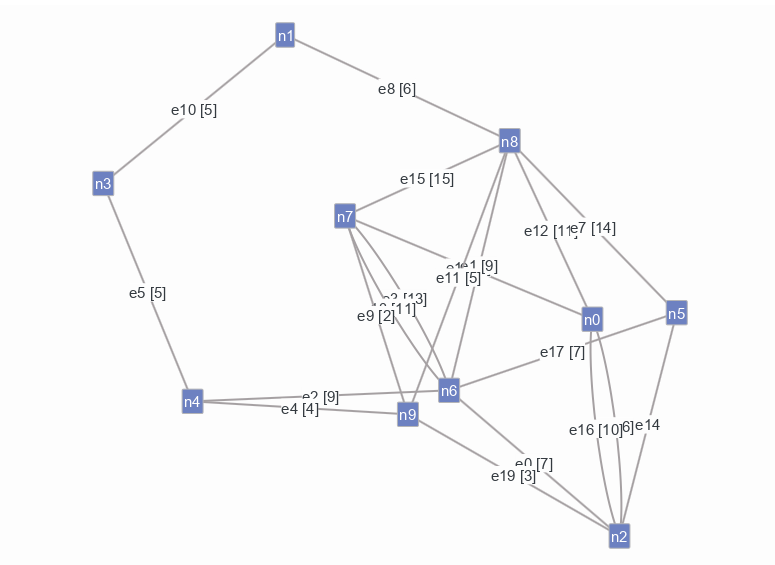


Abbildung 2 Graph nachdem restliche Kanten hinzugefügt worden sind

# GraphGenerator Test

Wir haben folgende Eigenschaften des Graphens getestet.

1. Anzahl der Übergebenden Kanten und Knoten sind im Graph zu finden
2. Knoten und Kanten Gewichtungen sind nicht höher als die Maximale Gewichtung
3. Graph ist zusammenhängend

Zu 1. Wir prüfen ob im Graph ein Knoten / Kante auf dem Index (Anzahl der Kanten / Knoten – 1) liegt.

Zu 2. Wir traversieren über den Graphen und schauen, ob die derzeitige Kante und Knoten0, Knoten1 eine Gewichtung < Maximale Gewichtung hat.

Zu 3. Wir wählen ein Knoten aus dem Graphen und schauen, ob alle weiteren Knoten des Graphens mit diesem verbunden sind.