

### Wegeproblem kürzester Weg fortsetzen

- noch einmal weiter austesten:  
Projekt Graphen Demoversion mit Schrittfunktion Dijkstra
- intensiv besprechen: `gallenbacher_dijkstra_objektorientiert.py`, darin insbesondere die Klasse Weg (Hinweis auf: Klasse Prioritätswarteschlange wird im Einzelprojekt zum ADT besprochen)
- nur Hinweis auf **P03-e Gallenbacher-Graph-Entwicklung.pdf** mit den Bildern aus dem Graphenprojekt
- Dazu **P03-b Gallenbacheralgorithmus-Py.pdf** mit dem Vergleich von Programm und der Algorithmusbeschreibung von Gallenbacher
- Anfang der Präsentation **P03-e Gallenbacher\_Kuerzester\_Weg.pdf** mit einer iterativen Lösung
- Hinweise:
  - Projektversion Dijkstra-Gallenbacher-Python mit der Startdatei gallenbacher\_dijkstra\_iterativ.py entspricht der Formulierung von Gallenbacher
  - In dem Projekt ist auch eine (teilweise) rekursive Lösung enthalten: gallenbacher\_dijkstra.py
  - Projektversion Dijkstra-Gallenbacher-rekursives-Py-Projekt entspricht der Präsentation **P03-c1 Dijkstra Funktionale Analyse\_Knoten-Py.pdf** und setzt die Modularisierung vollständig um
- prioritaetswarteschlange\_rekursiv.py / prioritaetswarteschlange\_iterativ.py / prioritaetswarteschlange\_objektorientiert.py besprechen und die zu Grunde liegende Datenstruktur (**A**bstakter **D**aten**T**yp) ADT Prioritätswarteschlange; einige Beispiele austesten!
- ~~Hinweis: Präsentation **P03-f Prioritaetswarteschlange-Py.pdf** oder **lohnt das auch(?) P03-f2 Prioritaetswarteschlange-Py-OO.pdf** und Projektversion Dijkstra-Gallenbacher\_objektorientiert, es ist die Basisversion für das Demoprojekt~~
- ~~Testen eines der Programme mit verschiedenen Start- und Zielorten.~~
  - ~~Bewertung der Qualität (nachholen?)~~
- Einordnung von Dijkstra: Präsentation **P03-g Dijkstra Typisierung.pdf**
- ~~Hinweis auf Material zu Dijkstra-Kanten-kleiner-Graph.zip~~  
→ Präsentation **P03-e3 Dijkstra-Schritte kleiner-Graph**

### Prim und Kruskal

- Hinweis: Zwei Algorithmen (**Prim** und **Kruskal**), im Material zu Anforderungen nur Kruskal
- Projekt Graphen Demoversion mit Schrittfunktion Kruskal zeigen und voll durchspielen
- Projektversion verwendet die Abschnitte zur OO Verwaltung einer Prioritätswarteschlange aus `gallenbacher_dijkstra_iterativ.py`, aber nur zum Sortieren!
- ~~Animation Kruskal-Algorithmus-kommentiert zeigen (auch für Prim, auch pdf) (nachholen)~~
- Präsentation **P04-b-1 Kruskal-Gallenbachergraph.pdf** (nachholen)
- Präsentation **P04-1 Algorithmus von Kruskal.pdf** (nachholen)

- Projekt ~~Algorithmus-von-Kruskal.zip~~, darin ~~Algorithmus-von-Kruskal.py~~ besprechen
- Projekt Graphen Demoversion mit Schrittfunktion Prim gezeigt
- Projektversion verwendet die Abschnitte zur OO Verwaltung der Prioritätswarteschlange aus `gallenbacher_dijkstra_iterativ.py` ersetzt Wege durch Kanten
- ~~Präsentation **P04-2 Von Dijkstra zu Prim.pdf** (nachholen)~~
- Präsentation **P04-2 Algorithmus von Prim.pdf** (nachholen)
- Präsentation **P04-b-2 Prim-Gallenbachergraph.pdf** (nachholen)
- ~~Gibt es eine Abhängigkeit vom Startknoten bei Prim?~~ (nachholen)