

☐ Gruppe M. Hava☐ Gruppe J. Heinzelreiter

Name: _____

Aufwand [h]: _____

☐ Gruppe P. Kulczycki

Feedback von: _____

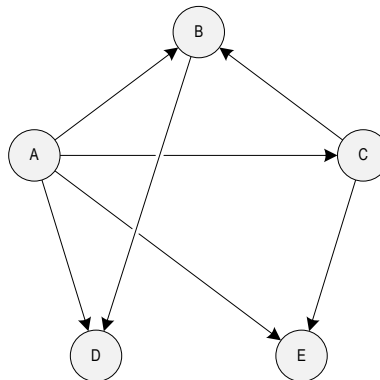
Beispiel	Lösungsidee (max. 100%)	Implement. (max. 100%)	Testen (max. 100%)
1 (25P + 40 P)			
2 (35 P)			

Beispiel 1: Abstrakter Datentyp „Gerichteter Graph“ (src/adt/)

Implementieren Sie den abstrakten Datentyp „Gerichteter Graph“ in zwei Varianten. Die eine Variante verwendet für den Graphen eine Nachbarschaftsmatrix (siehe dazu auch de.wikipedia.org/wiki/Adjazenzmatrix). Die andere Variante verwendet für den Graphen eine Nachbarschaftsliste (siehe dazu auch de.wikipedia.org/wiki/Adjazenzliste).

Beachten Sie die folgenden Anforderungen und Hinweise:

1. Beide Implementierungen haben identische Schnittstellen.
2. Beide Implementierungen verhalten sich – aus der Sicht eines Anwenders – absolut gleich.
3. Die „Payload“ eines Knotens ist eine beliebig lange, dynamisch allokierte Zeichenkette.
4. Kanten haben keine „Payload“ (also kein Gewicht etc.)
5. Die Anzahl der Knoten und Kanten, die der ADT aufnehmen kann, ist beliebig.
6. Knoten müssen dynamisch hinzugefügt und auch wieder gelöscht werden können. Wird ein Knoten gelöscht, so werden auch alle seine inzidenten Kanten gelöscht.
7. Kanten müssen hinzugefügt und auch wieder gelöscht werden können. Wird eine Kante hinzugefügt, so müssen seine inzidenten Knoten bereits Teil des ADTs sein.
8. Der ADT muss auf der Konsole entsprechend ausgegeben werden können.
9. Die Nachbarschaftsmatrix ist als dynamisch allokiertes (und damit eindimensionales) Feld aufzubauen.
10. Die Nachbarschaftsliste ist als eine Liste von Listen aufzubauen.
11. Führen Sie beide Implementierungen als separate C-Module (.c- und .h-Dateien) aus.
12. Implementieren Sie *ein* Testmodul (für beide Varianten) und testen Sie ausführlich.



Beispiel 2: Topologisches Sortieren (src/top/)

Implementieren Sie eine C-Funktion `topological_sort`, die die Knoten eines ADTs von Beispiel 1 topologisch sortiert auf der Konsole ausgibt. Diese Funktion können Sie als Teil des ADTs (in einer Variante Ihrer Wahl) ausführen.

Eine mögliche Ausgabe für den oben dargestellten Graphen wäre A,C,B,D,E. Siehe dazu auch en.wikipedia.org/wiki/Topological_sorting.