Terminplanung Termin 3 [22.02.24]

Wiederholung / nachholen

- iterative Version von fuelle-greedy ansehen: *fuelle-greedy-mit-for.py*
- Eingehen auf den Stack bei *fuelle-greedy-rekursiv-tiefe-liste-aufrufprotokoll.py*: Problem Endrekursion wird nicht erkannt!
- Rekursionsproblem bei Python zeigen mit Vergleich *rekursionsproblem.py* und *keinRekursionsproblem.scm*
- Hinweis auf Programmversion (und Begriff) zu flacher Liste in *fuelle-schrittweise-flache-Liste.zip*
  Es gibt aber zwei Varianten
  - Fassungsvermögen extern
  - Fassungsvermögen am Kopf der flachen Liste
- Präsentation ***P02-2 Binaerbaum_Container_100.pdf***

Fuelle viele

- Auf der Basis der gemeinsam erarbeiteten Version mit zwei Containern *fuelle-tiefensuche-zwei-Kurs.py*
  erweitern zu *fuelle-ts-viele.py* mit Iteration über die Liste
- Präsentation ***P02-d fuelle-viele.pdf***
- Projekt dazu *fuelle-viele-ts.scm*
- Laufzeitmessung mit *fuelle-viele-ts-laufzeit.scm* -> Berechnungsaufwand
- Unterschied in Laufzeit bei Erfolgsfall und Misserfolgsfall → warum?
  → dies ist eine **optimierte Tiefensuche**! (-> GT und HGT)

Breitensuche

- überhaupt behandeln?
- Präsentation ***P02-3 Breitensuche_80_Binaerbaum.pdf***
- Präsentation ***P02 Breitensuche-Praesentation.pdf***
- Präsentation ***P02a BS Entwicklung und Kohaesion.pdf*** weglassen?
- Präsentation ***P02b TS und BS im Vergleich.pdf***
- Projekte
  - *breitensuche-rucksack.py*, *breitensuche-rucksack-kurz.py* und *fuelle-breitensuche.py* als „ganz doofe“ Variante, die erst am Schluss testet
  - *fuelle-iterativ-GT.py*
- *fuelle-iterativ-HGT-tiefe-liste.py* sollte wohl besprochen und getestet werden
- *fuelle-iterativ-HGT.py*

Optimierung

- Präsentation ***P02-c Satisfizierung oder Optimierung.pdf***
- dazu Projekt *Optimierung-tiefensuche-tiefe-Liste.py* für die TS
  darin gezieltes Einsetzen von call-by-reference für den Parameter *bester*
- und *Optimierung.py* für die BS
- Hinweis aufPräsentation ***P03-e3 Dijkstra-Schritte kleiner-Graph***