

nachholen

- Hinweis auf Text **KI-00 Messung von Intelligenz.pdf**
- Hinweis auf GI-Artikel „Ohne Umwege“ von 09'24 zu Dijkstra
- Hinweis auf Arbeitsblätter **KI-00-d Arbeitsblatt schrittweise Erarbeitung.pdf** **KI-01-b1 Stuecke zum Ausschneiden.pdf** und **KI-01-b2 Stuecke groessengerecht.pdf**
- die einzelnen Schritte noch einmal erläutern mit der Präsentation **P02-1 Binaerbaum_Container_80.pdf**
- ~~Programm-Variante mit flacher Liste ansehen~~
- Präsentation **P02-2 Binaerbaum_Container_100.pdf**
- Text **KI-02-a Stack bei Tiefensuche.pdf**
- Text **KI-02-b Aufrufprotokoll_Tiefensuche.pdf**

Übungen zur Entwicklung mit Rekursion: (Bearbeitung in Gruppen !)

- Länge einer Liste (schrittweise: nur zwei Schritte!)
- sortieren-durch-Einfuegen.py (Kohärenz ansprechen! - schnittweise!) und zum Vergleich:
- Sortieren-durch-Einfuegen-iterativ.py zeigen
- ~~Angebot: Präsentationen aus OO zu Listen, Tupel, Strings usw~~

Fuelle viele

- nur als Lösungsidee für das Füllen von zwei Containern im Kurs angesprochen
- fuelle-zwei-tiefensuche-tiefe-liste.py verwenden (**nachholen**)
- ~~Präsentation P02-d fuelle-viele.pdf~~ (**nachholen**)
- ~~Projekt dazu fuelle-viele-ts.py~~ (**nachholen**)
- ~~Laufzeitmessung mit fuelle-viele-tiefensuche-tiefe-liste.py~~ (**nachholen**)
- ~~Unterschied in Laufzeit bei Erfolgsfall und Misserfolgsfall → warum?~~
→ dies ist eine **optimierte Tiefensuche!** (**nachholen**)

Breitensuche (überhaupt behandeln?)

- ~~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.py~~