

DSA Abgabe Blatt 00

Anton Sproll [], Omid Pourazar [], Jannik Becker [3597837]

16.04.2022

Aufgabe 1

a)

Eine Datenstruktur ist eine Ansammlung von mehreren beliebigen Datentypen/Objekten (allerdings pro Instanz immer der Gleiche Typ), die dort gelagert, verwaltet und verändert werden können.

Außerdem bieten sie Operationen an, die diese Verwaltung und Veränderung der in ihnen gespeicherten Daten ermöglichen.

b)

Ein String kann nur Buchstaben/Ziffern speichern und nicht beliebige Datentypen, deshalb ist ein String keine Datenstruktur. Außerdem bieten sie keine Operationen, die die Veränderung (zumindest in Java) ermöglichen, da Strings immutable sind.

c)

Ein Algorithmus ist eine Sequenz an Arbeitsschritten (z.B. Operationen/Instruktionen), die hintereinander ausgeführt werden und welche nach endlich vielen Schritten terminieren muss.

Ein Algorithmus wird i.d.R. möglichst allgemein und formal definiert, um eine Implementierung in unterschiedlichen Programmiersprachen zu ermöglichen.

d)

Manchmal können diese Abfolgen eines Algorithmus sehr komplex und verschachtelt werden, weshalb es in Fällen einfacher sein kann den Code in Pseudocode zu schreiben, um eine besseres, formaleres Verständnis zu bieten und die Übersicht über den Algorithmus zu verbessern.

Außerdem wird durch Pseudocode auf eine rein semantische Ebene abstrahiert, wodurch auch Programmierern ohne Kenntnisse der Syntax einer Programmiersprache ermöglicht wird die Funktion des Algorithmus zu begreifen und in einer ihnen bekannten Syntax/-Programmiersprache zu notieren.

Aufgabe 2 befindet sich dank LaTeX weiter unten

Aufgabe 3

a)

Wir bevorzugen die Liste, da diese dynamisch wachsen kann und so mit Leichtigkeit die Speicherkapazität verändert werden kann. Das ist nützlich, da wir nicht wissen, wie viele Elemente in dem gegebenen Programm verwaltet werden sollen.

Auch kann eine Aufgabe einfach in der Mitte der Liste hinzugefügt oder gelöscht werden, ohne dabei Speicherplatz zu verschwenden, da gelöschte Knoten keine “Lücken” hinterlassen. Bei Arrays wüssten wir nach dem löschen nicht, welches Feld zwischen gespeicherten Elementen potenziell frei ist oder müssten uns diese freien Felder in einer extra Datenstruktur merken, was deutlich aufwendiger zu implementieren wäre. Das Hinzufügen von Elementen in der Mitte eines Arrays wäre bei hingegen nur begrenzt möglich und nur durch eine aufwendigere und ineffizientere Implementierung wirklich fehlerfrei.

Außerdem kann das Hinzufügen von Elementen am Ende eines Arrays deutlich komplizierter werden, wenn das Array ausgelastet ist, da auch hier ein neues Array in einer geeigneten Größe erstellt und der Inhalt aus dem alten, ausgeschöpften Array in das neue kopiert werden muss. Das ist sehr ineffizient und unnötig aufwendig, wenn es doch Listen als simple Alternative gibt.

b)

Wenn eine festgelegte Menge an Daten zur Verfügung stehen, kann man auch einen Array zur Implementierung verwenden, da wir nun das Problem der Auslastung und dem Hinzufügen von neuen Elementen nicht haben.

Ein Beispiel für die Verwendung wäre, zumindest in C, Strings. Sie bestehen aus Arrays von Zeichen und Ganzzahlen von 0-9, und da Strings immutable sind, kann man keine Zeichen anhängen bzw. entfernen, wodurch keine dynamische Anpassung benötigt wurde. Das Auslesen von einzelnen Indizes ist dadurch sehr effizient.

Aufgabe 2

Algorithm 1 Spaghetti Pomodoro

```
1: Hole kleinen Topf aus Schrank
2: Stelle Topf auf Herd
3: Stelle Herd auf Stufe 7 von 10
4: Gebe Olivenöl hinzu
5: while Zutaten not kleingeschnitten do
6:   Schneide Tomaten und Zwiebeln
7: end while
8: Gebe Zwiebeln in kleinen Topf
9: while Zwiebeln not glasig do
10:   Warten und immer mal wieder rühren
11: end while
12: Gebe Tomaten in kleinen Topf
13: Gebe Gewürze hinzu
14: Stelle Deckel auf kleinen Topf
15: Stelle auf niedrigste Stufe und lasse köcheln
16:   1: Hole großen Topf aus Schrank
17:   2: while Topf not zu  $\frac{3}{4}$  mit Wasser gefüllt do
18:   3:   Fülle Wasser aus Wasserhahn in den Topf
19:   4: end while
20:   5: Schalte Herdplatte auf die höchste Stufe
21:   6: Stelle großen Topf auf Herdplatte
22:   7: Streue Salz ins Wasser
23:   8: while Wasser im Topf not am kochen do
24:   9:     Lasse Topf weitere Minute auf dem Herd stehen
25:   10: end while
26: 11: Hole und öffne Packung Nudeln
27: 12: Schütte Nudeln in Topf
28: 13: Schalte Herdplatte auf mittlere Stufe
29: 14: Stelle Deckel auf Topf
30: 15: while Nudel not al dente do
31: 16:   Lasse Nudeln 1 Minuten Kochen
32: 17: end while
33: 18: Schalte gesamten Herd aus
34: 19: Hole Sieb aus Schrank
35: 20: Stelle Sieb in Spüle
36: 21: Schütte Nudeln in Sieb
37: 22: if Nudeln is klebrig and Koch is not italiano then
38: 23:   Brause Nudeln mit Wasser ab
39: 24:   Durchmenge Nudeln
40: 25: end if
41: 26: Hole 4 Teller aus dem Schrank
42: 27: while Teller not angerichtet do
43: 28:   Fülle Teller mit Nudeln
44: 29:   Gebe Soße über Nudeln
45: 30: end while return Portionen
```
