2-3

- Hier handelt es sich offenbar um eine einfach verkette List, die, um das letzte Element zu ermitteln und ein Element am einzufügen, einmal komplett durchlaufen werden muss. Es liegt als beim Einfügen hinten eine O(n) Komplexität vor. Diese steigt also linear mit der Listengröße. Im Gegensatz dazu muss beim Einfügen vorne lediglich das Element eingefügt werden und die Referenz auf den übrigen Listenteil im neuen Element gespeichert werden. Die Komplexität O(1) ist als unabhängig von der Listengröße.

Verbesserungsvorschlag:

Speichert der Listenkopf auch das letzte Element der Liste, kann hinten änhlich effzient hinzugefügt werden wie vorne.

- Die Komplexität kann so von O(n) auf O(1) 'reduziert' werden. Das Einfügen am Ende ist nun ein Prozess, der unabhängig von der Listengröße immer gleich schnell abläuft.
- So lange der index nicht auf das letzte Element zeigt, kann dieser Vorschlag keine Laufzeitverbesserungen mit sich bringen. Die Liste kann immer noch lediglich in eine Richtung durchlaufen werden. n Schritte sind nötig, um nach dem n-ten Element einzufügen, oder die Daten des n-ten Elements zurückzugeben (O(n)).