

1. a)

Die verkettete Liste ist eine lineare Sammlung von Datenelementen, deren Reihenfolge nicht durch ihre physische Platzierung im Speicher gegeben ist. Stattdessen verweist jedes Element auf das nächste. Es ist eine Datenstruktur, die aus einer Sammlung von Knoten besteht, die zusammen eine Sequenz darstellen. In seiner einfachsten Form enthält jeder Knoten: Daten und einen Verweis (in anderen Worten: einen Link) auf den nächsten Knoten in der Sequenz. Diese Struktur ermöglicht ein effizientes Einfügen oder Entfernen von Elementen an jeder Position in der Sequenz während der Iteration. Komplexere Varianten fügen zusätzliche Verknüpfungen hinzu, die ein effizienteres Einfügen oder Entfernen von Knoten an beliebigen Positionen ermöglichen

3.

- Ein Array ist eine Datenstruktur, die eine Sammlung von ähnlichen Typen enthält, während die Linked List als eine nicht-primitive Datenstruktur betrachtet wird, die eine Sammlung von ungeordneten verknüpften Elementen enthält
- Arrays sind von fester Größe. Im Gegensatz dazu sind Linked Listen dynamisch und flexibel und können ihre Größe vergrößern und verkleinern.
- Im Array gehören die Elemente zu Indizes, d. h., wenn man in das vierte Element gelangen will, muss man den Variablennamen mit seinem Index oder seiner Position innerhalb der eckigen Klammer schreiben, während in einer verknüpften Liste muss man jedoch vom Kopf anfangen und sich bis zum vierten Element durcharbeiten.
- Operationen wie Einfügen und Löschen in Arrays verbrauchen viel Zeit. Auf der anderen Seite ist die Leistung dieser Operationen in verknüpften Listen schnell.

5.

In Bubble Sort werden im ersten Durchlauf  $n-1$  Vergleiche durchgeführt, im zweiten Durchlauf  $n-2$ , im dritten Durchlauf  $n-3$  und so weiter, daher ist die Zeitkomplexität von Bubble Sort  $O(n^2)$ . Die Platzkomplexität für Bubble Sort ist  $O(1)$ , da nur ein einziger zusätzlicher Speicherplatz benötigt wird, nämlich für die temporäre Variable.

Außerdem ist die Zeitkomplexität im besten Fall  $O(n)$ , nämlich dann, wenn die Liste bereits sortiert ist.

Im Folgenden sind die Zeit- und Raumkomplexität für den Bubble-Sort-Algorithmus aufgeführt.

Worst Case Time Complexity:  $O(n^2)$

Best Case Time Complexity:  $O(n)$

Average Time Complexity:  $O(n^2)$

Space Complexity:  $O(1)$

Selection sort sortiert ein Array, indem er wiederholt das kleinste Element (unter Berücksichtigung der aufsteigenden Reihenfolge) aus dem unsortierten Teil findet und an den Anfang setzt. Der Algorithmus verwaltet zwei Unterarrays in einem gegebenen Array.

In jeder Iteration der Auswahlsortierung wird das kleinste Element (unter Berücksichtigung der aufsteigenden Reihenfolge) aus dem unsortierten Teilarray ausgewählt und in das sortierte Teilarray verschoben.

Im Folgenden sind die Zeit- und Raumkomplexität für den Selection-Sort-Algorithmus aufgeführt.

Worst Case Time Complexity:  $O(n^2)$

Best Case Time Complexity:  $O(n^2)$

Average Time Complexity:  $O(n^2)$

Space Complexity:  $O(1)$