

L1: Dynamisches Array

Bekommen in L1

Abgabe in L3

Implementiere in C++ den gegebenen **Container (ADT)** mithilfe der gegebenen **Repräsentierung** und mit einem **dynamischen Vektor** als Datenstruktur. Den dynamischen Vektor muss man selber implementieren. Für die Implementierung dürft ihr keine Containers oder Datenstrukturen aus STL (oder aus anderen Bibliotheken) benutzen.

1. **ADT Matrix** – repräsentiert als schwachbesetzte Matrix (sparse), indem man ein dynamisches Array von Tupeln der Form (Zeile, Spalte, Wert) (Wert $\neq 0$) benutzt, wobei die Tupel in lexikographischen Reihenfolge nach (Zeile, Spalte) gespeichert werden.
2. **ADT Matrix** – repräsentiert als schwachbesetzte Matrix (sparse) in dem Compressed Column Storage Format (CCS) mithilfe von dynamischen Arrays
3. **ADT Matrix** – repräsentiert als schwachbesetzte Matrix (sparse) in dem Compressed Row Storage Format (CRS) mithilfe von dynamischen Arrays
4. **ADT Bag** – sequentielle Repräsentierung mithilfe von Paaren der Form (Element, Frequenz) in einem dynamischen Array (oder mit zwei dynamischen Arrays). Zum Beispiel, das Bag [5, 10, -1, 2, 3, 10, 5, 5, -5] wird folgendermaßen repräsentiert: [(5, 3), (10, 2), (-1, 1), (2, 1), (3, 1), (-5, 1)].
5. **ADT Bag** – repräsentiert als dynamisches Array von Frequenzen.
Zum Beispiel, das Bag [5, 10, -1, 2, 3, 10, 5, 5, -5] wird durch den Array
 $V = [1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 3, 0, 0, 0, 0, 2]$ repräsentiert, aufgebaut folgendermaßen:
 - Das Werteintervall [-5, 10] wird in das Intervall [0, 15] übersetzt
 - Auf der Position 0 in V speichert man die Frequenz des Wertes -5 (minimale Wert), auf der Position 1 in V speichert man die Frequenz des Wertes -4, ..., auf der Position 15 speichert man die Frequenz des Wertes 10 (maximale Wert)
6. **ADT Bag** – repräsentiert als ein dynamisches Array von eindeutigen Elementen (E) und ein dynamisches Array von Positionen (P) in E der Elemente des Bags
Zum Beispiel, das Bag [5, 10, -1, 2, 3, 10, 5, 5, -5] wird folgendermaßen repräsentiert:
 $U = [5, 10, -1, 2, 3, -5]$
 $P = [0, 1, 2, 3, 4, 1, 0, 0, 5]$
7. **ADT SortedBag** – mit Elementen vom Typ **TComp** repräsentiert mithilfe eines dynamischen Arrays von Paaren der Form (Element, Frequenz), sortiert mithilfe einer Relation auf den Elementen
8. **ADT SortedBag** – mit Elementen vom Typ **TComp** repräsentiert mithilfe eines dynamischen Arrays, sortiert mithilfe einer Relation auf den Elementen
9. **ADT SortedSet** – mit Elementen vom Typ **TComp** repräsentiert mithilfe eines dynamischen Arrays, sortiert mithilfe einer Relation auf den Elementen

- 10. **ADT Set** – repräsentiert mithilfe eines dynamischen Arrays von Elementen
- 11. **ADT Set** – repräsentiert mithilfe eines dynamischen Arrays von Boolean Werten (Bitarray)
- 12. **ADT Queue** – repräsentiert mithilfe eines zirkulären dynamischen Arrays von Elementen
- 13. **ADT Map** – repräsentiert mithilfe eines dynamischen Arrays von Paaren der Form (*key*, *value*)
- 14. **ADT SortedMap** – repräsentiert mithilfe eines dynamischen Arrays von Paaren der Form (*key*, *value*) und sortiert mithilfe einer Relation auf den Schlüsseln (*key*)