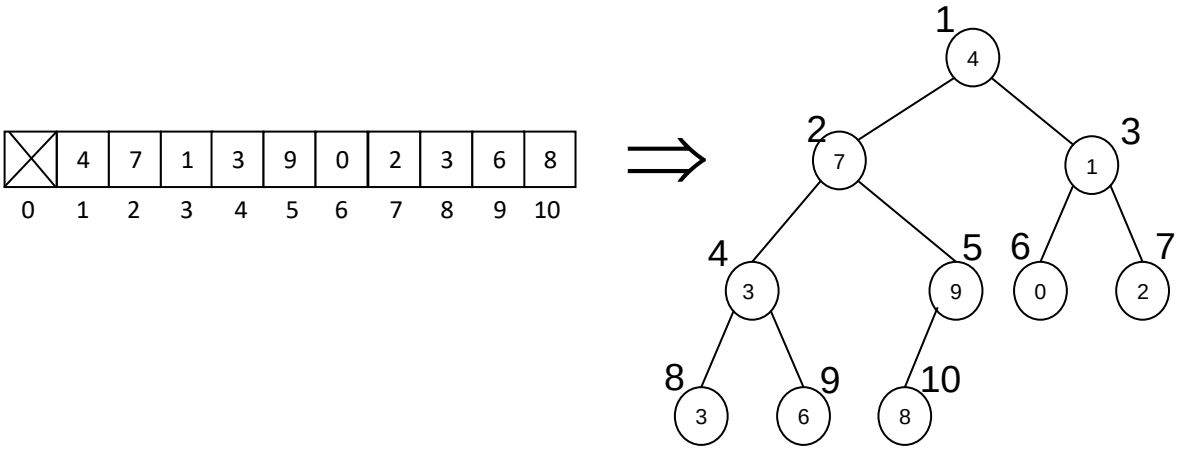
Elementare Algorithmen und Objektorientierte Programmierung

Übung 6

Auszuarbeiten bis 06.05.21

1. Darstellungsformen von Binären Bäumen (5 + 6 Punkte)

a) Gegeben ist ein vollständiger binärer Baum zur Speicherung von Ganzzahlenwerten. Dieser Baum in **sequentieller Darstellung** soll mit Hilfe der Funktion *Transform* in eine **verkettete Darstellung** umgeformt werden.




```
typedef int ItemType;
typedef struct Array {
      int nrOfValues;
      ItemType* pValues;
} Array_t;

typedef struct Node {
      ItemType value;
      struct Node* pLeft;
      struct Node* pRight;
} Node_t;
bool transform (Array_t* pTree, Node_t** ppRoot);
```

Die Daten für die sequentielle Darstellung sollen aus einer Datei eingelesen werden. (Geben Sie auch die Testdateien mit ab!)

b) Auf der verketteten Darstellung sind folgende Funktionen zu realisieren:

- `void destroy(Node_t** ppRoot);`
  Freigabe des gesamten Speicherplatzes des Baums d.h. den Speicherplatz den die Knoten belegen.
- `void printPreOrder(Node_t* pRoot)`
  Ausgabe des Baums am Bildschirm in PreOrder
- `void printInOrder(Node_t* pRoot)`
  Ausgabe des Baums am Bildschirm in InOrder
- `void printPostOrder(Node_t* pRoot)`
  Ausgabe des Baums am Bildschirm in PostOrder
- `int countLeaves(Node_t* pRoot)`
  Zählen der Blätter des Baums
- `int height(Node_t* pRoot)`
  Bestimmen der Höhe des Baums

2. Binärer Suchbaum (9 Punkte)

Entwickeln Sie die folgenden Funktionen für einen *Binären* Suchbaum, in dem Ganzzahlenwerte gespeichert:

- **Einfügen in einem Binären Suchbaum**
  Alle Nachfolger im *linken Unterbaum* haben *kleinere* oder *gleiche* Schlüsselwerte als der Knoten, alle Nachfolger im *rechten Unterbaum* haben *größere* Schlüsselwerte.
- **Bestimmen des Successors (Nachfolger) für einen gesuchten Wert**
  Der Nachfolger eine Knotens *K* ist der Knoten mit dem nächst größeren Integer-Werts
- **Bestimmen des Predecessors (Vorgänger) für einen gesuchten Wert**
  Der Vorgänger eine Knotens *K* ist der Knoten mit dem nächst kleineren Integer-Werts