

**Präsenzaufgabe 1.1**

Eine einfach-verkettete Liste sei durch die folgenden Arrays gegeben.

	A	NEXT
0	–	1
1	5	3
2	6	6
3	7	2
4	4	5
5	9	0
6	12	4
7		

- a) Stellen Sie die Zeigerstruktur der Liste dar.
- b) Geben Sie die Arrays nach Aufruf der Prozedur  $INSERT(Obj, p, q)$  mit  $Obj=42$ ,  $q=7$  und  $p=3$  an.
- c) Geben Sie die Arrays an, wenn nach der Änderung aus b) noch zusätzlich der Aufruf von  $DELETE(3)$  erfolgte.

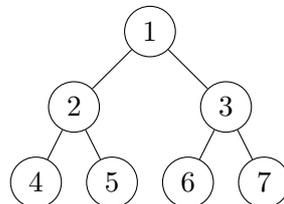
**Präsenzaufgabe 1.2**

Eine zweifach-verkettete Liste enthält die Elemente 5, 3, 6, 18, 11 in genau dieser Reihenfolge.

- a) Geben Sie eine mögliche Darstellung dieser Liste mit den Arrays A, NEXT und PREVIOUS an.
- b) Implementieren Sie die Prozedur  $INSERT(Obj, p, q)$ , die das Element  $Obj$  an Index  $q$  einfügt, wobei  $p$  den Index des vorherigen Listenelements darstellt.

**Präsenzaufgabe 1.3**

Gegeben sei folgender binärer Baum:



Geben Sie jeweils die Knotenreihenfolge in Prä-, Post- und Inorder an.

### Präsenzaufgabe 1.4

Gegeben seien drei Stacks  $S_1, S_2, S_3$ , auf die ausschließlich mit den folgenden Prozeduren zugegriffen werden kann:

- $\text{INSERT}_i(a)$  - Fügt das Element  $a$  zum Stack  $S_i$  hinzu
- $\text{GET}_i()$  - Gibt das oberste Element von  $S_i$  zurück (falls  $S_i$  nicht leer ist)
- $\text{DELETE}_i()$  - Löscht das oberste Element von  $S_i$
- $\text{SIZE}_i()$  - Gibt die Anzahl an Elementen, die in  $S_i$  gespeichert sind, zurück

Die Leerheit eines Stacks kann mit dem Test " $\text{IF SIZE}_i() = 0$ " abgefragt werden.

Neben den in der Vorlesung verwendeten Befehlen (z.B. Zuweisungen und  $\text{IF}$ ) sind auch  $\text{WHILE}$ -Schleifen erlaubt.

Der erste Stack enthält eine Folge von Zahlen, die anderen beiden Stacks seien anfangs leer.

- a) Entwerfen Sie einen Algorithmus, der die Reihenfolge der anfangs auf dem Stack  $S_1$  liegenden Elemente umkehrt.
- b) Entwerfen Sie einen Algorithmus, der die anfangs auf dem Stack  $S_1$  liegenden Elemente sortiert auf  $S_1$  ablegt.