



Informatik 1

11. Übung – Lösung

Aufgabe 1 (Ternärer Heap)

Ein ternärer Heap unterscheidet sich von dem in der Vorlesung eingeführten Heap dadurch, daß jeder Knoten drei Söhne haben kann. Dieser ist ebenfalls ein vollständiger (ternärer) Baum, der der Heap-Bedingung

$$\text{Inhalt}(\text{Vater}(i)) > \text{Inhalt}(i)$$

genügt.

- a) Wie repräsentieren Sie einen ternären Heap in einem Array? D.h. gegeben ein Knoten mit Nummer i , welche Nummern haben Vaterknoten, Knoten des linken, mittleren und rechten Sohnes?

$$\text{Vater } \lfloor (i + 1)/3 \rfloor, \text{ Söhne } 3i - 1, 3i, 3i + 1$$

- b) Welche Tiefe hat ein vollständiger ternärer Baum mit n Knoten höchstens?

$h \leq \log_3(2(n - 1) + 1)$, da im schlechtesten Fall (letzte Ebene enthält nur ein Element)

$$n = 1 + 3^1 + 3^2 + \dots + 3^{h-1} + 1 = \frac{3^h - 1}{3 - 1}$$

- c) Schreiben Sie die Funktion `delete_maximum` für einen ternären Heap.

```
function boolean delete_maximum(ternary_heap &h)
    integer l,m,r; // Indizes Söhne (linker,mittlerer,rechter)
    integer max; // maximaler Wert der Söhne
    integer t; // Arrayindex des max Sohns
    integer i; // Arrayindex aktuelles Element

    if h.n<1
```

```

then return false
fi

h.a[1]=h.a[h.n];
h.n=h.n-1;
i=1;
while 3i-1<=h.n do
  // finde max Sohn
  l=3i-1;
  m=3i;
  r=3i+1;

  max=h.a[l];
  t=l

  if m<=h.n && h.a[m]>max
  then t=m
      max=h.a[m]
  fi

  if r<=h.n && h.a[r]>max
  then t=r
      max=h.a[r]
  fi

  // vergleiche aktuelles Element mit max Sohn
  if h.a[i]<max
  then h.a[t]=h.a[i]
      h.a[i]=max
      i=t // tausche, Fortsetzung an dieser Stelle
  else return // keine weiteren Vertauschungen
  fi
od
return true;
end

```