



11. ÜBUNG ZUR VORLESUNG PROGRAMMIERUNG II, SS 2004

Aufgabe 1: Building Heap (10 Punkte)

Die Prozedur BUILD-HEAP

```
BUILD-HEAP(A)
  heap-size[A] = length[A]
  for i = length[A]/2 downto 1
    do HEAPIFY(A,i)
```

kann auch durch wiederholtes Aufrufen von HEAP-INSERT zum Einfügen der einzelnen Elemente in den Heap implementiert werden. Betrachte dazu folgenden Code:

```
BUILD-HEAP'(A)
  heap-size[A] = 1
  for i = 2 to length[A]
    do HEAP-INSERT(A,A[i])
```

- i) Erzeugen die Funktionen BUILD-HEAP aus der Vorlesung und BUILD-HEAP' immer den gleichen Heap, wenn sie mit demselben Eingabe-Array gestartet werden? Beweisen Sie dieses, oder widerlegen Sie es mit einem Gegenbeispiel!
- ii) Zeigen Sie, dass BUILD-HEAP' in worst case $\Theta(n \log(n))$ Zeit zum Bauen eines n -elementigen Heaps benötigt.

Aufgabe 2: Binärheaps (10 Punkte)

- i) Gegeben Sei das Array $A := [3,5,17,32,8,6,14,11,45,5]$. Handelt es sich bei A um einen binären Heap? Begründen Sie Ihre Aussage! Sollten Sie feststellen, dass A *keinen* binären Heap bildet, stellen Sie graphisch dar, wie die in der Vorlesung definierte Funktion `heapify` A in einen Heap verwandelt.
- ii) Gegeben Sei nun das Array $B := [4,0,10,7,15,8,2,17,5,6,12]$. Wandeln Sie B in einen Min-Heap um und stellen Sie graphisch dar, welche Operationen ausgeführt werden, wenn Sie das Minimum für diesen Heap löschen.

Aufgabe 3: Gerichtete Graphen (10 Punkte)

Unter dem *Quadrat* eines gerichteten Graphen $G = (V, E)$ versteht man den Graphen $G^2 = (V, E^2)$, wobei gilt: $(u, w) \in E^2 \Leftrightarrow \exists v \in V : (u, v) \in E \text{ und } (v, w) \in E$. Also wenn eine Kante zwischen u und v in E^2 liegt, muss G einen Pfad mit exakt zwei Kanten zwischen u und v enthalten. Beschreiben Sie einen effizienten Algorithmus zur Berechnung von G^2 von G sowohl für die Adjazenz-Liste als auch die Adjazenz-Matrix von G . Analysieren Sie die Laufzeit Ihrer Algorithmen!

Aufgabe 4: Euler-Circuit (10 Punkte)

Unter der *Eulerisierung* eines Graphen G versteht man den Prozess des Hinzufügens von duplizierten Kanten zu G , so dass der resultierende Graph einen Euler-Circuit hat.

- i) Beschreiben Sie einen effizienten Algorithmus zur Eulerisierung eines Graphen G , und beweisen Sie seine Korrektheit.
- ii) Eulerisieren Sie den Graphen G aus Abb.(b).

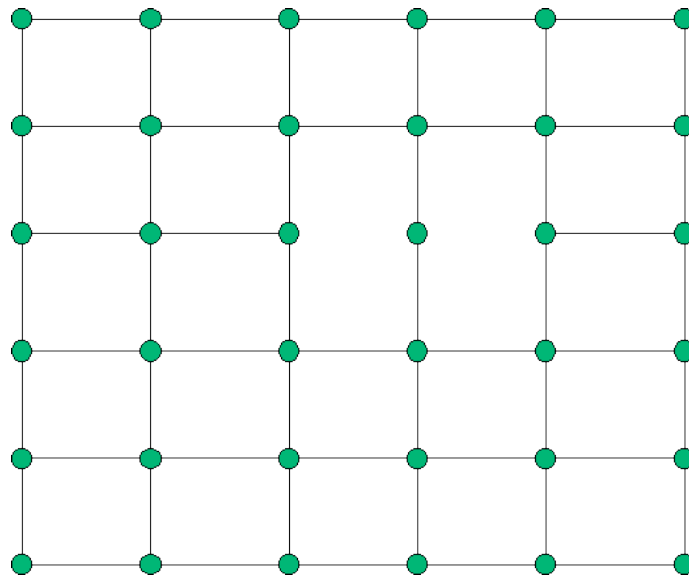


Abbildung 1: Der Graph G von Aufgabe 4 (iii)