Klausur zur Vorlesung
Algorithmen und Datenstrukturen

Prof. Dr. Matthias Rarey

WS 2011/12

1. Termin, 14. Februar 2012

Aufgabe	1	2	3	4	5	6	$\sum$
Punkte	20	20	20	20	20	20	120
Ergebnis							

Name: ______________________________

Matrikelnummer: ______________________________

Bearbeitungshinweise:
Bitte sorgfältig durchlesen und beachten!

- Die Bearbeitungsdauer der Klausur beträgt 120 Minuten
- Zum Bestehen der Klausur sind 50 Punkte hinreichend
- Überprüfen Sie bitte sofort nach Erhalt die Vollständigkeit der Unterlagen (21 Seiten)
- Bitte lassen Sie die Klausur zusammengeheftet. Notieren Sie alle Lösungen direkt auf den Aufgabenblättern
- Verwenden Sie kein eigenes Papier, sondern nur die zur Verfügung gestellten Blätter
- Bitte schreiben Sie mit blauem oder schwarzem Schreiber – Bleistift und Rotstift sind nicht erlaubt.
- Schreiben Sie bitte auf jedes Blatt Ihren Namen und Ihre Matrikelnummer
- Falls notwendig benutzen Sie die Rückseite des jeweiligen Aufgabenblattes für Notizen und Entwürfe
- Als Hilfsmittel ist nur ein mit Ihrem Namen und Ihrer Matrikelnummer versehenes, jeweils zur Hälfte zweiseitig handbeschriebenes DIN A4 Blatt erlaubt
- Bitte legen Sie Ihren Studentenausweis und einen Lichtbildausweis auf den Tisch.
- Schalten Sie Ihr Handy aus!

Ich bestätige hiermit, dass ich alle notwendigen Voraussetzungen laut der für mich gültigen Prüfungsordnung erbracht habe. Für den Fall, dass die Voraussetzungen zum Zeitpunkt der Klausurteilnahme nicht erfüllt waren, nehme ich zur Kenntnis, dass die Klausur nicht gewertet wird.

Datum: ________________ Unterschrift: ______________________________

Name: ______________________ Matrikelnummer: ____________

Aufgabe 1: (20 Punkte)

Bei allen Multiple-Choice Fragen ist genau ein Kreuz zu setzen. Sollte mehr als ein Kreuz gesetzt sein, wird die Teilaufgabe mit 0 Punkten gewertet.

1. Gegeben sind die Funktionen

$$n^2 \qquad n! \qquad n^{2n} \qquad \sqrt[3]{n} \qquad \log(n)$$

   Welche der folgenden Aussagen gilt? (1 Punkt)

   - [ ] Alle Funktionen sind polynomiell.
   - [ ] $n!$ ist die am stärksten wachsende Funktion
   - [ ] $\sqrt[3]{n} = O(\log(n))$
   - [ ] $n^2 + o(n^2) = \Theta(n^2)$

2. Seien $f(n), g(n)$ Funktionen. Weiter sei h definiert durch $h(n) = \max(f(n), g(n))$. Gilt dann (2 Punkte)

   - [ ] $h(n) = O(f(n))$
   - [ ] $h(n) = O(f(n) + g(n))$
   - [ ] $h(n) = \omega(f(n) + g(n))$
   - [ ] $h(n) = O(f(n) - g(n))$

3. Beschreiben Sie die Datenstruktur Schlange, die sich durch die folgende Sequenz von Operationen ergibt: (1 Punkt)

   ENQUEUE(15), REAR(), ENQUEUE(7), ENQUEUE(8), DEQUEUE(), ENQUEUE(9), DEQUEUE(), HEAD(), ENQUEUE(10)

4. Wir betrachten einen Baum T mit Höhe h (die Wurzel hat die Höhe 0) und n Knoten. Die inneren Knoten von T haben maximal drei Kinder. Welche der folgenden Aussagen ist wahr? (2 Punkte)

   - [ ] h ist immer $\log_3(n)$
   - [ ] T hat mindestens n Kanten
   - [ ] T hat immer mehr Blätter als innere Knoten
   - [ ] T hat mindestens $n - 3^h$ innere Knoten.

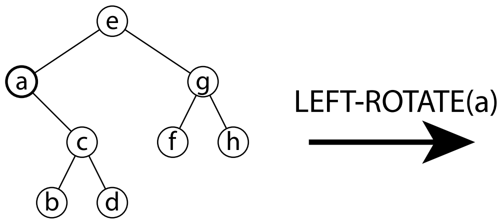
5. Sei $G = (V, E)$ ein zusammenhängender, ungerichteter Graph mit der Kantengewichtsfunktion w. Angenommen $w(e) = c$, c konstant, für alle Kanten, welche Laufzeit lässt sich dann zur Berechnung des minimalen Spannbaumes in G erreichen? (2 Punkte)

- [ ] $O(|V| \log |E|)$
- [ ] $O(|V| + |E|)$
- [ ] $O(|E| \log |V|)$
- [ ] $O(|V|^2)$

6. Fügen Sie den Wert 12 in die folgende Hashtabelle mit offener Addressierung und linearer Sondierung ein. Als Hashfunktion wird $h(x) = x \mod 7$ benutzt. (2 Punkte)

0	1	2	3	4	5	6
		9			5	13

7. Führen Sie eine Linksrotation auf den Knoten a aus. (2 Punkte)



8. Seien $L, L' \in P$. Welche der folgenden Aussagen ist **falsch**? (2 Punkte)

- [ ] Wenn $L \in NP$ gilt, dann folgt $P = NP$
- [ ] $\forall L'' \in NPC : L \leq_p L''$
- [ ] $L \cap L' \in P$
- [ ] $L \cup L' \in P$

9. Welches ist die untere Schranke für die Anzahl der Vergleichsoperationen bei vergleichsbasierten Sortieralgorithmen? (2 Punkte)

   - [ ] $\Omega(n)$
   - [ ] $\Omega(\log(n))$
   - [ ] $\Omega(n \log(n))$
   - [ ] $\Omega(n^2)$

10. In einem minimalen Spannbaum eines Graphen mit n Knoten (2 Punkte)

    - [ ] kann die Kante mit dem höchsten Gewicht enthalten sein
    - [ ] darf die Kante mit dem kleinsten Gewicht nicht enthalten sein
    - [ ] können mehr als $n-1$ Kanten enthalten sein
    - [ ] muss die Summe der Kantengewichte durch 2 teilbar sein

11. In einem Rot-Schwarz-Baum gilt (ohne Betrachtung der Wächter) für alle roten Knoten r (2 Punkte)

    - [ ] der Vor-Vorgänger von r ist rot, falls der Elter von r nicht die Wurzel ist
    - [ ] der Bruder Knoten von r ist rot
    - [ ] Wenn r kein Blatt ist, dann hat r zwei Kinder
    - [ ] der Onkel von r ist schwarz

Name: ______________________ Matrikelnummer: ____________

Aufgabe 2: (20 Punkte)

1. Betrachten Sie die folgenden Code-Segmente. Geben Sie für jedes Segment eine möglichst dichte Schranke für die asymptotische Laufzeit im O-Kalkül in Abhängigkeit von N, bzw. $N = \text{length}[A]$ an. Bei rekursiven Algorithmen formulieren Sie zuvor die Rekurrenzgleichung. (Teil-Arrays werden immer als Kopie übergeben, das Kopieren kostet keine Zeit.) (14 Punkte)

```
ALG1()
  for i = 0 to N
      do for j ← N downto 1
             do sum ← sum + j
```

$O(\qquad\qquad)$

```
ALG2()
  for i = 1 to N
      do j ← N
         while j > 1
             do sum ← sum + j
                j ← j/2
```

$O(\qquad\qquad)$

```
ALG3()
  while N > 1
      do for j ← 1 to N
             do sum ← sum + j
         N ← N/2
```

$O(\qquad\qquad)$

```
ALG4(A)
  middle ← length[A]/2
  sum ← sum + A[1]
  if length[A] > 2
     then ALG4(A[1 .. middle])
          ALG4(A[middle + 1 .. length[A]])
```

$T(N) = \begin{cases} \\ \\ \end{cases}$

$O(\qquad\qquad)$

```
ALG5(A)
  for i ← 1 to length[A]
      do sum ← sum + A[i]^2
  if length[A] > 3
     then ALG5(A[1 .. length[A]/3])
          ALG5(A[length[A]/3 .. 2 · length[A]/3])
          ALG5(A[2 · length[A]/3 .. length[A]])
```

$T(N) = \begin{cases} \\ \\ \end{cases}$

$O(\qquad\qquad)$

Name: ______________________________ Matrikelnummer: _______________

ALG6(A)
  for $i \leftarrow 1$ **to** length$[A]$
    do $sum \leftarrow sum + A[i]$
  if length$[A] > 1$
    then ALG6($A[1\,..\,\text{length}[A]-2]$)

$T(N) = \begin{cases} \\ \\ \end{cases}$

$O($ $)$

ALG7(A)
  $sum \leftarrow sum + 1$
  if length$[A] > 1$
    then ALG7($A[1\,..\,\text{length}[A]-2]$)
      ALG7($A[2\,..\,\text{length}[A]-1]$)

$T(N) = \begin{cases} \\ \\ \end{cases}$

$O($ $)$

2. Bestimmen Sie die asymptotische Laufzeit des folgenden Algorithmus mit der Substitutionsmethode, geben Sie Ihren Lösungsweg an: (6 Punkte)

```
ALG8(A, N, n)
  if n ≥ 2
    then for i ← 1 to n
              do SWAP(A[i], A[r])
                 ALG8(A, N, n − 1)
                 SWAP(A[r], A[i])
    else  sum ← 0
          for i ← 1 to N
              do s ← s + A[i]
```

Die Funktion wird aufgerufen mit: ALG8(A, length[A], length[A])

Hinweise:

a) Verwenden Sie folgende Notation: $[n]_i = n \cdot (n-1) \cdot \ldots \cdot (n-i)$

b) Verwenden Sie die Abschätzung $[n]_i \leq n!$, wenn dies nicht zu einer Erhöhung der Laufzeitschranke im O-Kalkül führt.

Aufgabe 3: (20 Punkte)

1. Gegeben sei die bekannte Partitionsfunktion aus dem Quicksort-Algorithmus

```
PARTITION(A, l, r)
1  ▷ l - linkes Blockende, r - rechtes Blockende, A[r] - Pivot
2  i ← l − 1
3  for j ← l to r − 1
4       do if A[j] ≤ A[r]
5            then i ← i + 1
6                 SWAP(A[i], A[j])
7  SWAP(A[i + 1], A[r])
8  return (i + 1)
```

Dabei gehen wir in dieser Aufgabe von Arrays mit 0-basiertem Index aus. Sie befinden sich in Zeile 4 der PARTITION-Funktion mit $l = 0, r = 6$ und $j = 3, i = 0$. Werten Sie alle Schritte bis zum Erreichen der Zeile 8 aus. Protokollieren Sie den Speicherzustand jeweils beim Erreichen der Zeile 4 und der Zeile 8. (6 Punkte)

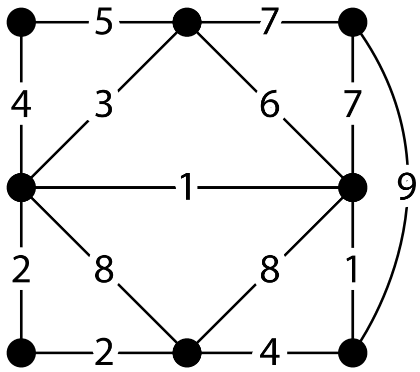
Zeile 4, A:	3	8	6	4	6	4	5	$j = 3, i = 0$
Zeile 4, A:								$j = 4, i =$
Zeile 4, A:								$j = 5, i =$
Zeile 8, A:								$j = 5, i =$

2. Ist Quicksort mit der obigen Partition Funktion ein stabiles Sortierverfahren? Begründen Sie Ihre Antwort und zeigen Sie entweder, dass Partition stabil ist oder schlagen Sie eine Modifikation von Quicksort vor, welche diesen stabilisiert. (6 Punkte)

3. Gegeben sei ein Array der Länge N, welches ausschließlich Elemente aus der Menge $0, 1, 2$ enthält. Entwerfen Sie einen Algorithmus, der die Elemente in $O(n)$ Zeit sortiert. Der Algorithmus soll vergleichs-/austauschbasiert und *in-place* arbeiten, was bedeutet, dass er maximal $O(1)$ zusätzlichen Speicher nutzen darf. Skizzieren Sie Ihre Idee mit Pseudo-Code und einem beschreibenden Text. (8 Punkte)

Name: ______________________ Matrikelnummer: ______________

Aufgabe 4: (20 Punkte)

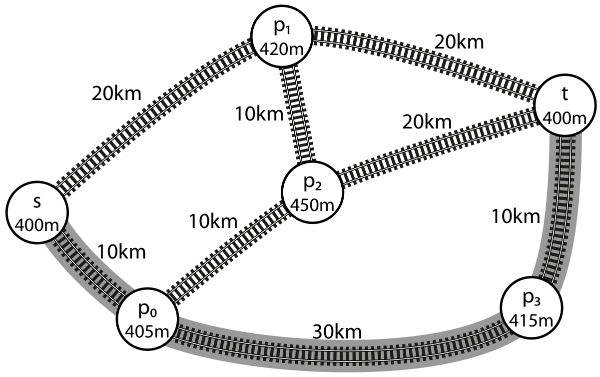


1. Zeichnen Sie in den oben angegebenen Graphen einen minimalen Spannbaum ein. Ist dieser Spannbaum eindeutig? Begründen Sie Ihre Antwort. (4 Punkte)

2. Gegeben sei ein Graph $G = (V, E)$. Entwerfen Sie einen rekursiven Algorithmus $\textsc{Label}(V, E, u, \text{'label'})$, der einen Knoten u und alle von u aus erreichbaren Knoten mit der Markierung 'label' versieht. Der Algorithmus soll die Laufzeit $O(|V| + |E|)$ haben. Beschreiben Sie Ihren Algorithmus und geben Sie Pseudo-Code für die Funktion Label an. (8 Punkte)

3. In einem Graphen $G = (V, E)$ mit Kantengewichten bezeichnen wir eine Kante e eines minimalen Spannbaums T als kritisch, wenn das Entfernen von e das Gewicht des minimalen Spannbaums in G vergrößert. Entwickeln Sie einen Algorithmus, der in $O(|V| + |E|)$ Zeit entscheidet, ob e kritisch ist. Nutzen Sie dabei die Funktion Label aus der vorausgehenden Aufgabe als Hilfsfunktion. Beschreiben Sie kurz Ihren Lösungsweg und geben Sie den Algorithmus im Pseudo-Code an. Argumentieren Sie außerdem in wenigen Stichpunkten, warum Ihr Algorithmus korrekt ist.
(8 Punkte)

Aufgabe 5: (20 Punkte)



Eine neue Eisenbahnlinie soll gebaut werden und Sie wurden mit ihrer Planung beauftragt. Zur Verfügung steht Ihnen eine topographische Landkarte mit möglichen Streckenverlaufspunkten p_i. Auf der Karte sind die Distanzen in km zwischen den Punkten eingetragen sowie die Höhe der Streckenverlaufspunkte über Normalnull in Metern. Ihr Auftrag lautet, eine Bahnlinie zwischen zwei gegebenen Punkten s, t zu bestimmen, auf welcher ein Zug möglichst wenig Energie umsetzt. Die umgesetzte Energie bestimmt sich dabei wie folgt:

- 1 Energieeinheit pro km Strecke
- 2 Energieeinheiten pro Anstieg um einen Meter
- -1 Energieeinheit pro Abfall um einen Meter

1. Wieviel Energie wird auf dem in der Karte grau markierten Pfad von s nach t (über p_0 und p_3) umgesetzt? Gibt es einen energetisch günstigeren Pfad? (4 Punkte)

2. Modellieren Sie das Problem so, dass Sie einen aus der Vorlesung bekannten Algorithmus benutzen können. Beschreiben Sie Ihre Modellierung und nennen Sie den Algorithmus, der angewendet werden kann. (8 Punkte)

3. Durch Einsatz einer moderneren Lok ändert sich der Energieverbrauch pro Anstieg um einen Meter auf 1 Energieeinheit. Zeigen Sie, dass nun im allgemeinen Fall für zwei beliebige Knoten s, t gilt: der Pfad von s nach t mit den wenigsten Streckenkilometern entspricht dem Pfad mit dem minimalem Energieverbrauch. (8 Punkte)

Name: ______________________________ Matrikelnummer: ______________

Aufgabe 6: (20 Punkte)

Paul hat einen Kasten mit N Bauklötzen, die Klötze k_i haben jeweils die Maße b_i, t_i und h_i (Breite / Tiefe / Höhe). Paul fragt sich, ob er mit den Klötzen einen Turm bauen kann, der exakt genau so hoch ist wie sein Schreibtisch (Schreibtischhöhe H). Dabei kann jeder Klotz beliebig gestapelt werden.

1. Zeigen Sie durch Reduktion auf ein Ihnen bekanntes Problem, dass Pauls Problem NP-vollständig ist. (6 Punkte)

Name: ______________________ Matrikelnummer: ____________

2. Entwickeln Sie ein Backtracking-Algorithmus zur Lösung von Pauls Problem. Skizzieren Sie in wenigen Stichpunkten Ihren Lösungsweg und geben Sie den Algorithmus in Pseudo-Code an. (6 Punkte)

3. Paul möchte nun den höchsten Turm bauen, der noch unter seinen Schreibtisch passt. Geben Sie Schranken an, die verwendet werden können um den Backtracking-Algorithmus in einen Branch&Bound Algorithmus umzuwandeln. (4 Punkte)

4. Paul stellt beim Vermessen seiner N Klötze fest, dass alle eine Grundfläche von 1×1 cm haben und außerdem entweder 6, 3 oder 2 cm hoch sind. Ist Pauls Problem unter dieser Randbedingung noch NP-vollständig? Begründen Sie Ihre Antwort. (4 Punkte)