Themen: Divide & Conquer-Algorithmen
Experimentelle Analyse

Alle Dokumente bitte im Algorithmik-Wiki im PDF-Format ablegen!

Aufgabe 1 (Divide & Conquer)

a) Schreiben Sie MergeSort(A, p, r) rekursiv als Divide&Conquer-Algorithmus im Pseudocode. Unterscheiden Sie im Algorithmus deutlich die drei Schritte *Divide*, *Conquer* und *Combine*.

Hinweise: A = (A[1], A[2], ..., A[n]) ist die zu sortierende Folge, p ist der Index der ersten Elements und r der Index des letzten Elements der Folge A.

Gegeben ist eine Prozedur Merge(A, p, q, r), die zwei Folgen (A[p], ..., A[q]) und (A[q+1], ..., A[r]) zu einer sortierten Folge (A[p], ..., A[r]) mischt.

b) Identifizieren Sie im Pseudocode die Schritte *Divide*, *Conquer* und *Combine* und ermitteln Sie für jeden dieser Schritte die Zeitkomplexität.

c) Stellen Sie mit den Überlegungen aus a) eine Differenzgleichung für MergeSort auf.

d) Lösen Sie die Differenzgleichung mit Hilfe des Mastertheorems.

Aufgabe 2 (**k**-**k**leinstes **E**lement (kkE))

Gegeben: Eine Folge von n Datensätzen

Gesucht: Die Nummer des Datensatzes, der in der nach aufsteigenden Schlüsseln sortierten Folge an k-ter Stelle stehen wird.

Lösen Sie das Problem auf zwei verschiedene Art und Weisen:

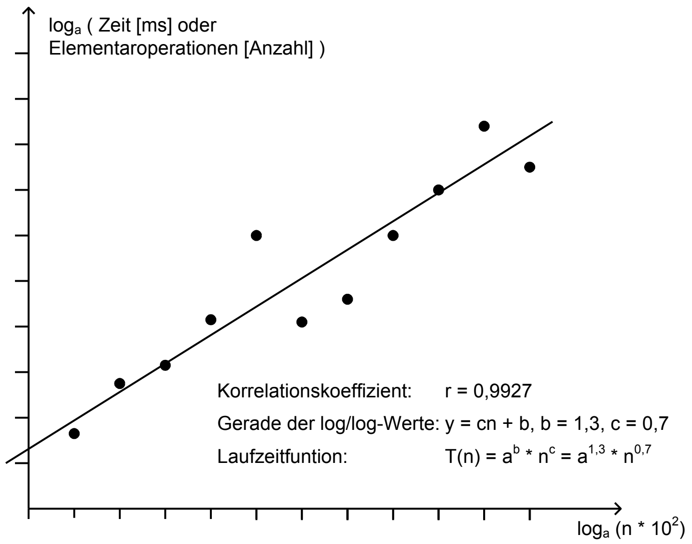
- Mit einem Algorithmus Ihrer Wahl
- Mit dem D&C-Algorithmus **randomizedQuickSelect (rQS)**

a) Formulieren Sie beide Algorithmen im Pseudocode. Erklären Sie die Idee der Algorithmen.

b) Programmieren Sie die Algorithmen in Java.

c) Führen Sie für beide Algorithmen eine asymptotische und eine experimentelle Analyse durch. Verwenden Sie für die experimentellen Analysen den Powertest. Geben Sie den Korrelationskoeffizienten r, die log/log-Gerade und die Laufzeitfunktion t(n) an.

Vergleichen Sie die Ergebnisse der experimentellen mit denen der asymptotischen Analyse und prüfen Sie, ob die Resultate konsistent sind. Stellen Sie die Ergebnisse der Experimente nach folgenden bzw. dem Muster unter Punkt 3 der Hinweise dar.



- Logarithmierte (Basis a) Mess- bzw. Zählwerte

—— Durch lineare Regression ermittelte Gerade y = cn + b

a: Basis des Logarithmus
b: Schnittpunkt der Geraden mit der y-Achse
c: Steigung der Geraden

Hinweise zu den Aufgaben

1. Der rQS-Algorithmus

Gegeben sei eine unsortierte Folge S von n vergleichbaren Elementen und eine Integerzahl k ∈ [1, n]. Wähle zufällig ein Element x aus S und benutze dieses als Pivot-Element, um S in die drei Subsequenzen L, E und G zu teilen. L speichert alle Elemente von S, die kleiner als x, E alle die gleich x und G alle die größer als x sind. Dieser Schritt heißt **Decrease**- oder auch **Prune**-Schritt (prune = abschneiden). Anhand des Vergleichs von k mit der Kardinalität |L|, |E| und |G| der drei Mengen wird entschieden, mit welchen Parameterwerten für S und k der Algorithmus **randomizedQuickSelect** rekursiv aufgerufen wird. Der Pseudocode der Algorithmus-Idee sieht wie folgt aus:

```
Algorithmus randomizedQuickSelect (S, k):
  Input:  Sequenz S von n vergleichbaren Elementen und Integer k ∈ [1, n]
  Output: Das k-kleinste Element von S
  if n = 1 then
    return das (erste) Element von S
  wähle ein Element x zufällig aus S aus
  entferne alle Elemente aus S und speichere sie in den Folgen:
     -L, speichert alle Elemente aus S kleiner als x
     -E, speichert alle Elemente aus S gleich x
     -G, speichert alle Elemente aus S größer als x

  if k ≤ |L| then           ... ?
  else if k ≤ |L| + |E| then    ... ?
  else  randomizedQuickSelect (G, k - |L| - |E|)
```

2. Durchführung von experimentellen Analysen

Experimentelle Analysen bestehen aus

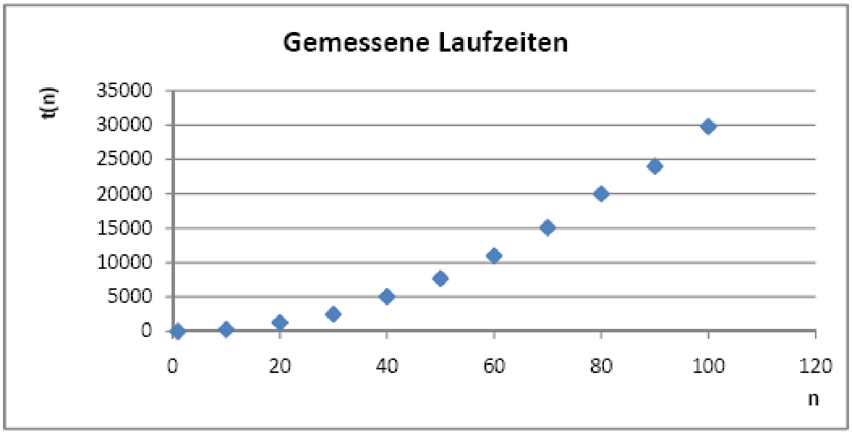
- sorgfältiger Planung,
- Durchführung,
- Auswertung und
- **Dokumentation**.

Folgende Dinge sind zu beachten:

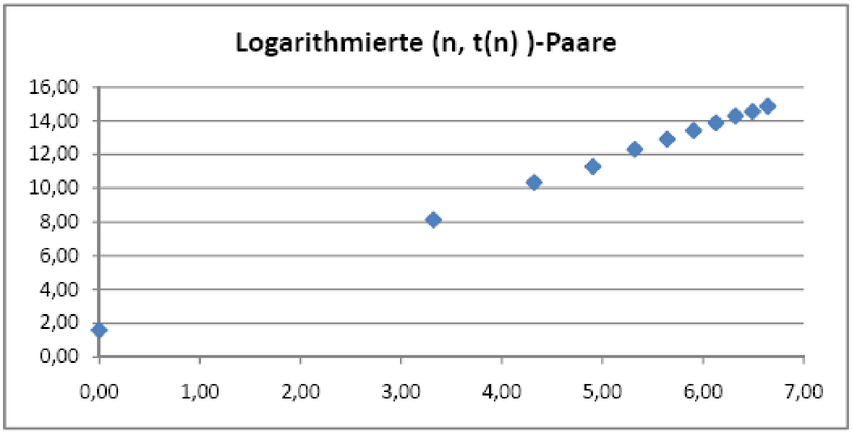
- Soll die experimentelle Analyse durch eine Zeitmessung erfolgen oder durch Zählen von Elementaroperationen? Wenn von einem Algorithmus kein Quellcode verfügbar ist, kommt nur eine Zeitmessung in Frage. Steht Quellcode zur Verfügung, müssen typische Elementaroperationen bzw. die typische Elementaroperation ermittelt und gezählt werden.
- Für aussagekräftige Ergebnisse müssen repräsentative Daten erzeugt werden. Beim kkE-Problem müssen z.B. Folgen von jeweils n Datensätzen mit einem guten **Zufallszahlengenerator** generiert werden. Oft ist es sinnvoll, die Problemgröße n in größeren Schritten zu variieren, z.B. n = 100, 200, 300, ..., weil sich sonst die gemessenen Laufzeiten für verschiedene n nur unwesentlich unterscheiden.
- Bei der Messung von Laufzeiten ist zu berücksichtigen, dass andere Prozesse auf dem Rechner die Messungen beeinflussen können. Um dieses Problem zu mildern, sollten die Experimente mehrmals wiederholt und dann der Mittelwert bestimmt werden.
- Stellen Sie die Ergebnisse in einer (n, t)-Tabelle dar. Beim **Powertest** müssen die (n, t)-Paare **logarithmiert** werden, um eine Geradengleichung zu erhalten.
- Durch lineare Regression sind die logarithmierten (n, t)-Paare auf Korrelation zu prüfen. Ganz wichtig: **Korrelationskoeffizient** r berechnen und auszugeben!
- Die **Ergebnisse** der experimentellen Analyse sollen in **tabellarischer** und **grafischer Form** ausgegeben werden. Für die grafische Ausgabe eignet sich z.B. die Java-Bibliothek *Ptplot*, http://ptolemy.berkeley.edu/java/ptplot/index.htm.
- Dokumentieren Sie die Experimente und erklären Sie,
  - wie die Daten erzeugt wurden,
  - welche Messergebnisse erzielt wurden (tabellarisch und grafisch),
  - welche Laufzeit für den untersuchten Algorithmus ermittelt wurde und
  - ob die Ergebnisse mit der theoretischen Analyse übereinstimmen

3. Beispiel für die Darstellung der Experiment-Ergebnisse

n	t(n)
1	3
10	280
20	1300
30	2500
40	5050
50	7700
60	11000
70	15100
80	20000
90	24000
100	29800



$\log_2(n)$	$\log_2(t(n))$
0,00	1,58
3,32	8,13
4,32	10,34
4,91	11,29
5,32	12,30
5,64	12,91
5,91	13,43
6,13	13,88
6,32	14,29
6,49	14,55
6,64	14,86



Schnittpunkt: 1,58
Steigung: 2,0067
Korr.-Koeff. r = 0,9998
T(n) = $2^{1,58} * n^{2,01}$

Korrelationskoeffizienten nicht vergessen!!

n	T(n)
1,0000	3,00
10,0000	304,63
20,0000	1224,16
30,0000	2761,79
40,0000	4919,26
50,0000	7697,76
60,0000	11098,22
70,0000	15121,42
80,0000	19767,98
90,0000	25038,46
100,0000	30933,35