



Abgabe: 20.-22. Mai 2008 beim jeweiligen Tutor

Praktikum Grundlagen der Programmierung

Themen: Suchen und Sortieren

Wiederholung: Felder, Rekursion und Iteration

Aufgabe 20 (Ü) **Ein einfacher Suchalgorithmus**

Um ein Element in einem Feld zu suchen, kann man folgendermaßen vorgehen:

Durchlaufe das Feld und vergleiche jedes Element mit dem gesuchten Element.

- Unter welche Bedingungen ist die Anzahl der erforderlichen Vergleiche minimal/maximal?
- Implementieren Sie eine Methode, die als Parameter ein `int[]` und eine zu suchende Zahl erhält. Ist die Zahl im Feld enthalten, soll der Index der Zahl im Feld zurückgeliefert werden, ansonsten -1.

Aufgabe 21 (Ü) **Ein schneller Suchalgorithmus**

Ist ein Feld bereits sortiert, so kann man wie folgt darin suchen:

Um festzustellen, ob ein gegebenes Element in einem aufsteigend sortierten Feld enthalten ist, wird es zuerst mit dem Element auf der mittleren Position des Feldes verglichen. Wenn das gesuchte Element kleiner ist als das Mittlere, muss es in der linken Hälfte des Feldes liegen, andernfalls in der Rechten. Diese Methode wendet man nun rekursiv auf das Teilfeld an.

- Wieviele Vergleiche sind maximal erforderlich, um eine gegebene Zahl in einem sortierten Feld mit n Zahlen zu finden?
- Implementieren Sie eine Methode, die als Parameter ein `int[]` und das zu suchende Element erhält und bei erfolgreicher Suche dessen Index zurückgibt, ansonsten -1.

Aufgabe 22 (Ü) Ein einfacher Sortieralgorithmus

Einer der einfachsten Sortieralgorithmen läuft wie folgt ab:

Finde zuerst das kleinste Element im Feld und tausche es gegen das an der ersten Stelle befindliche Element aus, finde danach das zweitkleinste Element und tausche es gegen das an zweiter Stelle befindliche Element aus. Fahre in dieser Weise fort, bis das gesamte Feld sortiert ist.

Implementieren Sie diesen Algorithmus.

Aufgabe 23 (H) Sortieren von Spielkarten (4 Punkte)

Ein weiteres einfaches Verfahren, das Menschen oft beim Kartenspiel anwenden, um ihre Karten zu sortieren, läuft wie folgt ab:

Betrachte die Elemente eines nach dem anderen und füge jedes an seinen richtigen Platz zwischen den bereits betrachteten ein (wobei diese sortiert bleiben). Das gerade betrachtete Element wird eingefügt, indem die größeren Elemente einfach um eine Position nach rechts bewegt werden und das Element dann auf dem freigewordenen Platz eingefügt wird.

Implementieren Sie diesen Algorithmus.

Aufgabe 24 (H) Ein langsamer Sortieralgorithmus (4 Punkte)

Ein elementares, wenn auch langsames Sortierverfahren läuft wie folgt ab:

Durchlaufe das Feld immer wieder und vertausche jedesmal, wenn es notwendig ist, benachbarte Elemente. Wenn bei einem Durchlauf kein Austausch mehr stattfindet, ist das Feld sortiert.

- a) Implementieren Sie den beschriebenen Algorithmus.
- b) Modifizieren Sie Ihre Implementierung so, dass nach dem Sortiervorgang jedes Element nur noch einmal im Rückgabe-Array enthalten ist.

Aufgabe 25 (H) Felder und Mengen (6 Punkte)

Implementieren Sie eine Klasse `FelderMengen.java` mit Methoden, die als Parameter jeweils zwei `int[]` erhalten und folgendes zurückgeben:

- a) Ein `int[]`, welches alle Elemente aus den übergebenen Feldern in absteigender Sortierung enthält. Im Rückgabefeld dürfen keine doppelten Elemente enthalten sein.
- b) Ein `int[]`, welches alle Elemente die in beiden Feldern vorkommen, genau einmal enthält.
- c) Ein `int[]`, welche alle Elemente genau einmal enthält, die nur in einem der beiden Felder vorkommen.