Algorithmen und Datenstrukturen

(Th. Ottmann und P. Widmayer)

Folien: Vorsortieren

Autor: Sven Schuierer

Institut für Informatik
Georges-Köhler-Allee
Albert-Ludwigs-Universität Freiburg

1 Überblick

Überblick

Untere Schranke für allgemeine Sortierverfahren

Entscheidungsbäume

Sortieren vorsortierter Daten

2 Untere Schranke für allgemeine Sortierverfahren

Satz

Zum Sortieren einer Folge von n Schlüsseln mit einem allgemeinen Sortierverfahren sind im Worst-case ebenso wie im Mittel wenigstens $\Omega(n \log n)$ Vergleichsoperationen zwischen zwei Schlüsseln erforderlich.

Modellierung von allgemeinen Sortierverfahren: Entscheidungsbäume.

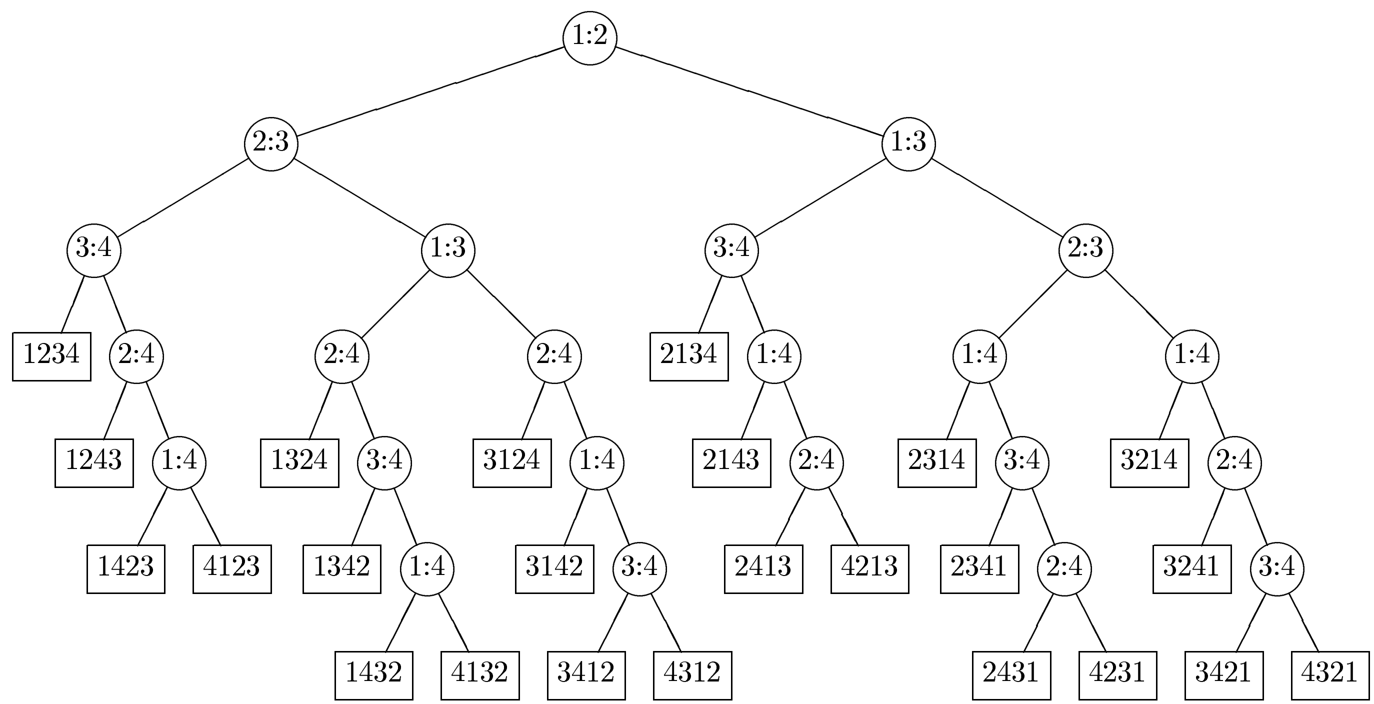
3 Entscheidungsbäume

Sortierverfahren A

Entscheidungsbaum $T_{A,n}$ zur Modellierung von des Ablaufs von A auf Folgen der Länge n enthält:

- für jede der $n!$ Permutationen ein Blatt
- innere Knoten repäsentieren eine Vergleichsoperation und haben zwei Söhne
- Weg W von der Wurzel zu einem Blatt v:
  — die Vergleiche an den Knoten von W identifizieren die Permutation π_v von v
  — entsprechen den von A durchgeführten Vergleichen, falls die Eingabe π_v ist

Beispiel:
Sortieren durch Einfügen für Folge
$F = < k_1, k_2, k_3, k_4 >$ von 4 Schlüsseln


1:2
2:3
1:3
3:4
1:3
3:4
2:3
1234
2:4
2:4
2:4
2134
1:4
1:4
1:4
1243
1:4
1324
3:4
3124
1:4
2143
2:4
2314
3:4
3214
2:4
1423
4123
1342
1:4
3142
3:4
2413
4213
2341
2:4
3241
3:4
1432
4132
3412
4312
2431
4231
3421
4321

4 Sortieren vorsortierter Daten

Spezielle Eingaben treten häufiger auf, Daten sind vorsortiert

Vorsortierungsmaße für eine Folge F

Anzahl der Inversionen

$$inv(F) = \left|\{(i,j) \mid 1 \le j < i \le n, \quad k_j > k_i\}\right|$$

Beispiel: 15 2 43 17 4 8 47

F aufsteigend sortiert: $inv(F) = 0$

F absteigend sortiert: $inv(F) = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2}$

inv mißt die globale Vorsortierung

Beispiel:

$n/2 + 1, \ldots, n, 1, \ldots, n/2$

Anzahl aufsteigend sortierter Teilfolgen

$$\begin{aligned} runs(F) &= |\{i \mid 1 \leq i \leq n, \quad k_i > k_{i+1}\}| + 1 \\ &= \text{Anzahl aufsteigend sortierter Teilfolgen} \end{aligned}$$

Beispiel: F : 15 2 43 17 4 8 47

F aufsteigend sortiert: $runs(F) = 1$
F absteigend sortiert: $runs(F) = n$

$runs$ mißt die lokale Vorsortierung

Längste aufsteigende Teilfolge

$$\begin{aligned} las(F) &= \text{Länge der longest ascending subsequence}(F) \\ &= \max\{t \mid \exists 1 \le i_1 < \cdots < i_t \le n, \\ &\qquad\qquad k_{i_1} \le k_{i_2} \le \cdots \le k_{i_t}\} \end{aligned}$$

$1 \le las(F) \le n$

Beispiel:

F: $15, 2, 43, 17, 4, 8, 47,$ $\quad las(F) =$

$rem(F) = n - las(F)$

F aufsteigend: $rem(F) = 0$

absteigend: $rem(F) = n - 1$

Optimale Nutzung der Vorsortierung

m-optimales, allgemeines Sortierverfahren

Ziel: für jeden Wert m_0 von m soll nur die für Folgen dieses Vorsortiertheitsgrades nötige Schrittzahl verwendet werden.

Untere Schranke für Anzahl der nötigen Schlüsselvergleiche C_{m_0}?

Anzahl der Blätter im Entscheidungsbaum mit $m(F) \leq m_0$

$$|\{F \mid m(F) \leq m_0\}|.$$

Definition

Ein Sortierverfahren A heißt m-optimal, falls es eine Konstante c gibt, so daß für alle n und alle Folgen F mit Länge n die Zeit $T_A(F)$ zum Sortieren von F mit A wie folgt beschränkt ist:

$$T_A(F) \leq c \cdot (n + \log |\{F' \mid m(F') \leq m(F)\}|).$$