PD Dr. rer. nat. habil. Sven Groppe

Übungen zur Vorlesung

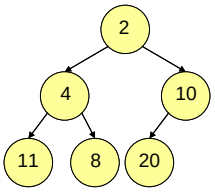
Semantic Web

WS 2013/2014

Übung 10 – Indexkonstruktion

Aufgabe 1:

Wie kann man einen kompletten Binärbaum wie zum Beispiel



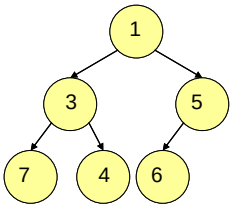
in ein eindimensionales Feld (*Array*) ablegen? Entwickle Formeln zur Bestimmung des linken und rechten Kindes sowie des Elternknotens eines an der Position *i* im Feld abgelegten Knotens.

Aufgabe 2:

Füge 4, 7, 3, 5, 6 und 2 in einen leeren Heap ein und zeichne den Heap nach jeder Einfügeoperation.

Aufgabe 3:

Entferne das kleinste Element vom folgenden Heap



bis der Heap leer ist und zeichne den Heap nach jedem Schritt.

Aufgabe 4:

Sortiere die Sequenz 7, 3, 6, 1, 9, 15, 2, 14, 5, 4, 11 und 8 bei Verwendung von

a) Externem Merge Sort, wobei 6 Elemente im Hauptspeicher sortiert werden können,

b) Replacement Selection mit Verwendung eines Heaps der Größe 6,

c) Externem Chunks Merge Sort bei Verwendung eines 3-Chunks Heap und 6 Elemente passen in den Hauptspeicher und

d) Distribution Sort bei Verwendung der Verteilungsschlüssel 3, 6, 9 und 12.

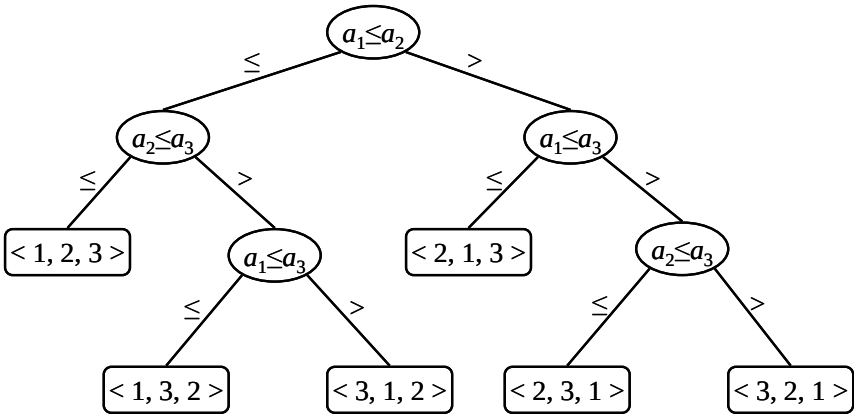
Visualisiere jeden Schritt während des Sortierens.

Aufgabe 5:

Sei n die Anzahl der zu sortierenden Elemente $a_1, \ldots, a_n$. In dieser Aufgabe wollen wir die untere Schranke $\Omega(n*\log(n))$ für den schlechtesten Fall für die Klasse V von Sortierverfahren beweisen, die nur über Vergleiche zwischen zwei Elementen Informationen der relativen Ordnung innerhalb der Eingabesequenz verwenden. Zu dieser Klasse von Sortieralgorithmen gehören zum Beispiel Bubblesort, Heapsort, Quicksort, Shellsort, Mergesort, Direktes Aussuchen und Direktes Einfügen, aber nicht Bucketsort.

Ohne Einschränkung der Allgemeinheit seien alle zu sortierenden Elemente verschieden und alle Vergleiche haben die Form $a_i \leq a_j$, wenn a_i und a_j zu sortierende Elemente sind.

Die notwendigen Vergleiche für beliebige zu sortierende Elemente $a_1, \ldots, a_n$ von Sortierverfahren aus V können abstrakt durch *Entscheidungsbäume* repräsentiert werden. Zur Ermittlung einer unteren Schranke für Sortierverfahren aus V werden dabei andere Aspekte wie die zur Programmkontrolle und zur Datenbewegung ignoriert. Die unten stehende Abbildung zeigt zum Beispiel einen Entscheidungsbaum für das Sortierverfahren Direktes Einfügen für 3 zu sortierende Elemente.



Jeder innere Knoten von Entscheidungsbäumen repräsentiert dabei einen Vergleich $a_i \leq a_j$ zwischen den zu sortierenden Elementen. Jeder Blattknoten von Entscheidungsbäumen repräsentiert eine Permutation $< \pi(1), \ldots, \pi(n) >$ der zu sortierenden Elemente $a_1, \ldots, a_n$. Die Ausführung eines Sortieralgorithmus entspricht dabei einem Pfad von der Wurzel bis zu einem Blattknoten des Entscheidungsbaumes. Bei jedem inneren Knoten wird ein Vergleich $a_i \leq a_j$ vollzogen. Der linke Teilbaum des aktuellen inneren Knotens enthält die nachfolgenden Vergleiche, falls $a_i \leq a_j$ gilt, und der rechte Teilbaum enthält die nachfolgenden Vergleiche, falls $a_i > a_j$ gilt. Bei Erreichen eines Blattknotens hat das Sortierverfahren die Ordnung $a_{\pi(1)} \leq a_{\pi(2)} \leq \ldots \leq a_{\pi(n)}$ bestimmt.

Beweisen Sie, dass jeder Entscheidungsbaum, der n Elemente sortiert, die Höhe $\Omega(n*\log(n))$ hat und damit jedes Sortierverfahren aus der Klasse V die Worst-Case-Laufzeit $\Omega(n*\log(n))$ besitzt.

Bemerkung: Es gilt $n! > (n/e)^n$ (Approximation von Stirling), wobei e die Eulersche Zahl 2,71828… repräsentiert.