Algorithmen und Datenstrukturen
Sommersemester 2024

Alexandros Paraskevaidis
Elias Gestrich

3. Übungsblatt

Aufgabe 1 - MultiSort

a) $T(\text{MultiSort}(\text{int}[]A)) \in \Theta(n(n+n+n)) \in \Theta(n^2)$, da die äußere Schleife in $\Theta(n-1)$ liegt und die inneren `for`-Schleifen in $\Theta(n)$ mal laufen und MySort in $\Theta(n)$ Zeit braucht um ein Element in eine sortierte Liste einzufügen.

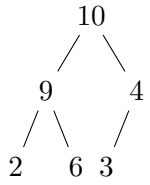
b) $T(\text{MultiSort}(\text{int}[]A)) \in \Theta(n(n+n^2+n)) \in \Theta(n^3)$, da im Best-Case die Laufzeit von QuickSort $\Theta(n^2)$, da $B[1 \dots i-1]$ schon sortiert, also $i-1$-mal jedes Element verglichen wird und das einzufügende Element schon an der richtigen Stelle steht.

c) $T(\text{MultiSort}(\text{int}[]A)) \in \Theta(n(n+n\log n+n)) \in \Theta(n^2 \log n)$, da im erwarteten Fall Quicksort in $\Theta(n \log n)$ liegt.

Aufgabe 2 - Heap Heap Hurra

insert(WT 4):
$[4,-,-,-,-,-]$
insert(WT 2):
$[4,2,-,-,-,-]$
insert(WT 9):
$[4,2,9,-,-,-]$
$[9,2,4,-,-,-]$
insert(WT 6):
$[9,2,4,6,-,-]$
$[9,6,4,2,-,-]$

insert(WT 10):
$[9,6,4,2,10,-]$
$[9,10,4,2,6,-]$
$[10,9,4,2,6,-]$
insert(WT 3):
$[10,9,4,2,6,3]$



extractMax():
$[3,9,4,2,6,-]$
$[9,3,4,2,6,-]$
$[9,6,4,2,3,-]$
extractMax():
$[3,6,4,2,-,-]$
$[6,3,4,2,-,-]$
extractMax():
$[2,3,4,-,-,-]$
$[4,3,2,-,-,-]$
extractMax():
$[2,3,-,-,-,-]$
$[3,2,-,-,-,-]$

Aufgabe 3 - Qual der Wahl

a) Für einen Sortieralgorithmus mit einer Worst-Case-Laufzeit von $\Theta(n \log n)$: Bucketsort: $\Theta(n \log n)$ aus Vorlesung.
CountingSort: $\Theta(n^4)$.
RadixSort: $\Theta(s(n+d)) = \Theta(\log_d(n^4)(n+d)) = \Theta(4 \log_d(n)(n+d)) \in \Theta(n \log(n))$, also unabhängig von der Basis

b) Man kann mit BucketSort die Zahlen rausfischen, die größer sind als $10^6 \cdot n$, diese mithilfe eines beliebigen Algorithmuses sortieren und für die restlichen CountingSort verwenden.

Aufgabe 4 - Basiswörter

man gibt jeden Buchstaben eine Zahl von 1 bis 26 mit $a = 1, b = 2, \dots, z = 26$ und dem Leerzeichen als 0, wobei Wörter mit drei Buchstaben ein Leerzeichen am Ende eingefügt wird, und sortiert dann die Wörter als wären sie vierstellige 27-äre Zahlen:

ice	car	map	pen	crab	frog	duck	lion	nest	boat
[9, 3, 5, 0]	[3, 1, 18, 0]	[13, 1, 16, 0]	[16, 5, 14, 0]	[3, 18, 1, 2]	[6, 18, 15, 7]	[4, 21, 3, 11]	[12, 9, 15, 14]	[14, 5, 19, 20]	[2, 15, 1, 20]
crab	boat	duck	ice	pen	frog	lion	map	car	nest
[3, 18, 1, 2]	[2, 15, 1, 20]	[4, 21, 3, 11]	[9, 3, 5, 0]	[16, 5, 14, 0]	[6, 18, 15, 7]	[12, 9, 15, 14]	[13, 1, 16, 0]	[3, 1, 18, 0]	[14, 5, 19, 20]
map	car	ice	pen	nest	lion	boat	crab	frog	duck
[13, 1, 16, 0]	[3, 1, 18, 0]	[9, 3, 5, 0]	[16, 5, 14, 0]	[14, 5, 19, 20]	[12, 9, 15, 14]	[2, 15, 1, 20]	[3, 18, 1, 2]	[6, 18, 15, 7]	[4, 21, 3, 11]
boat	car	crab	duck	frog	ice	lion	map	nest	pen
[2, 15, 1, 20]	[3, 1, 18, 0]	[3, 18, 1, 2]	[4, 21, 3, 11]	[6, 18, 15, 7]	[9, 3, 5, 0]	[12, 9, 15, 14]	[13, 1, 16, 0]	[14, 5, 19, 20]	[16, 5, 14, 0]