Übungsaufgabe 05 zu Algorithms and Optimizations

14.11.2015

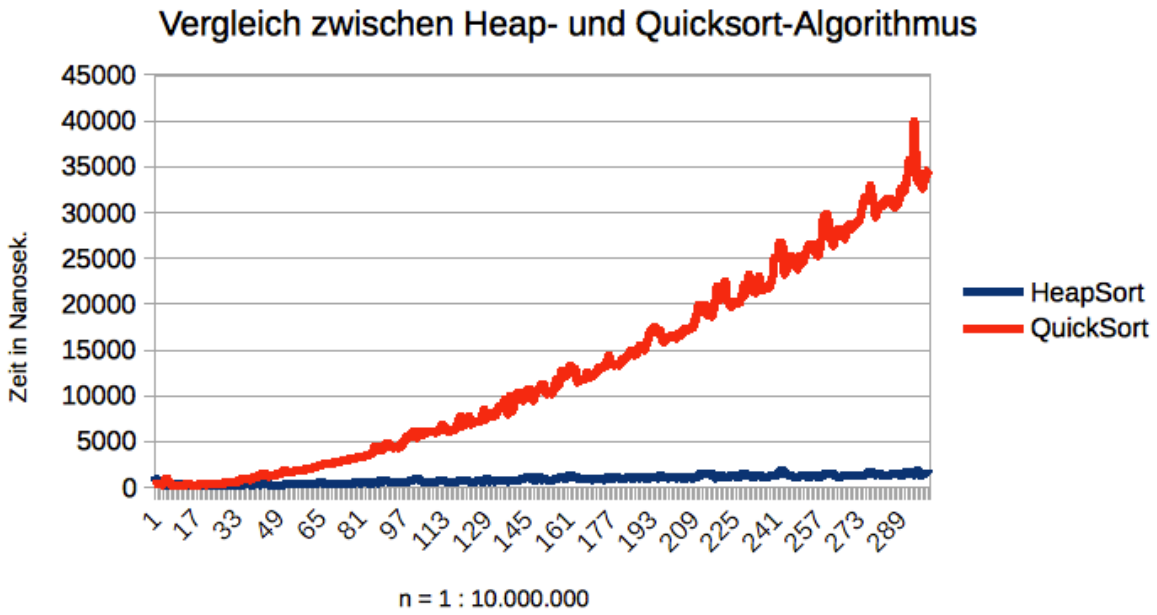
Gruppenmitglieder:

Tu Tran Thi Ngoc: 537318 André Vallentin: 527538 Jakob Warnow: 531600

Aufgabe 1:

Implementieren Sie HeapSort. Vergleichen Sie die Laufzeit Ihrer Implementierung, angewendet auf zufällige Listen, mit der Laufzeit der Implementierung von Quicksort aus der Lehrveranstaltung (oder Ihrer eigenen Implementierung aus Hausaufgabe 4).

Für jeden Durchlauf (n) wurde die zu sortierende Liste mit 10.000.000 Zufallswerten erweitert. Jede einzelne Sortiervorgang wurde 1.000x durchgeführt um einen ansprechenden Mittelwert zu finden.



Anhand der Grafik lässt sich deutlich ablesen, dass der HeapSort-Algorithmus die zufällige Liste deutlich schneller sortiert und logarithmisch wächst.
Beim Quicksort-Algorithmus steigt die Ausführungszeit mit längeren Listen fast quadratisch an.

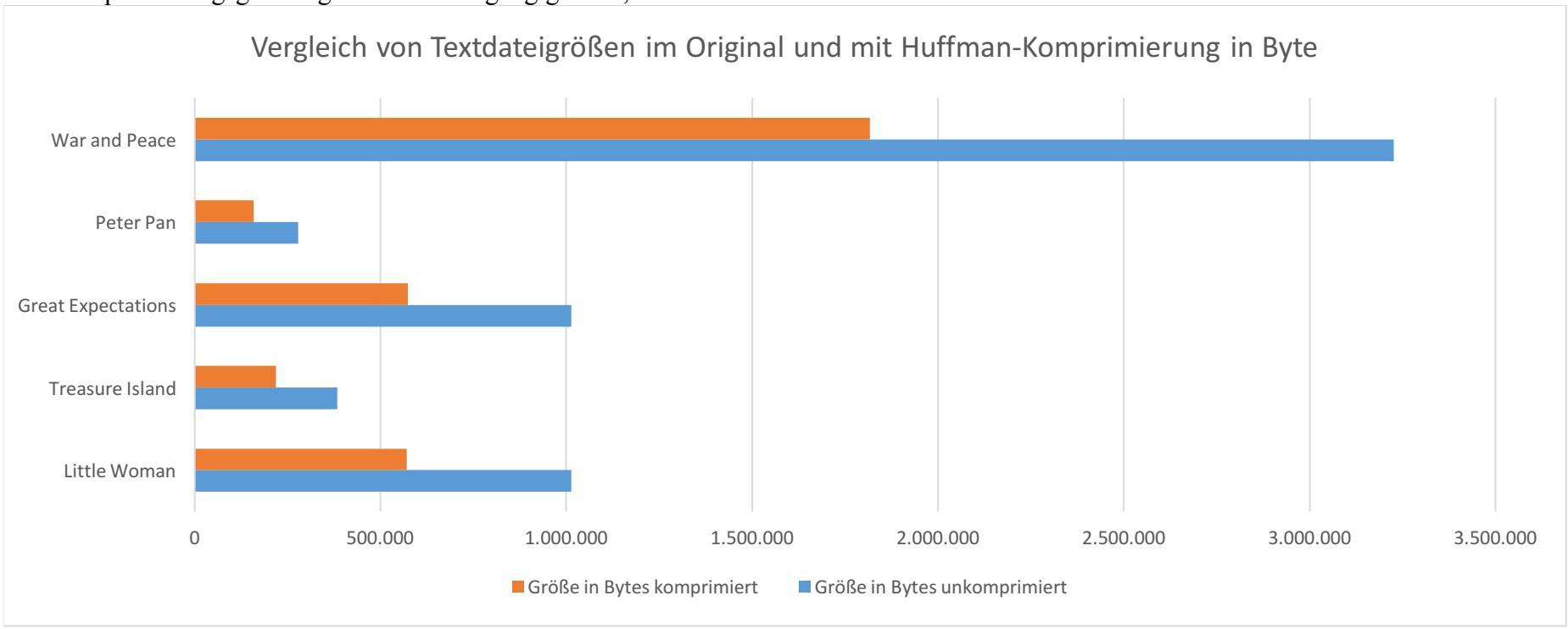
Aufgabe 2:

Schreiben Sie ein Programm, mit dem Sie ein Textfile Huffman-codieren können. Welche Reduktion der Filegröße erreichen Sie damit?

Hier ein Vergleich zwischen fünf klassischen Werken der Literatur:

Romantitel	Größe in Bytes unkomprimiert	Größe in Bytes komprimiert	Komprimierung in Prozent
Little Woman	1.013.662	571.124	43,66
Treasure Island	383.673	218.177	43,13
Great Expectations	1.012.626	573.407	43,37
Peter Pan	278.144	158.814	42,90
War and Peace	3.226.620	1.816.597	43,70

Die Werke wurden von der Plattform: http://www.gutenberg.org/ bezogen und als reiner ASCII-Text eingelesen, komprimiert und dekomprimiert.
Der Komprimierungsgrad liegt bei ca. durchgängig 43 %, worin die meisten Zeichensätze der ASCII-Tabelle im Huffman-Baum enthalten sind.



Die Huffman-Code Implementierung legen wir Ihnen im Zip-File mit bei.

Anbei ein **Enkodierungsbeispiel** vom Roman „Peter Pan“:

\n	11110
	001
!	0111101010
"	0001100
#	0111100000111010
$	01111000001110011
%	011110000011100001
'	010111011
(	0101110001000
)	0101110001010

*	0001101110010
,	011101
-	111110010
.	0000100
/	0001101000110
0	111110011000
1	11111001101
2	0101110001011
3	1111100110010
4	1111100110011

5	00011010001111
6	00011010001110
7	01111000001101
8	01111000001100
9	0101110001001
:	000110100000
;	0001101001
=	01111000001110010
?	0101110000
@	01111000001110001

A	0001101101
B	0111101000
C	00011010111
D	0111100011
E	0111100010
F	01011100011
G	00011011101
H	011110010
I	01111011
J	1111100111

K	00011011100111
L	00011011110
M	0001101100
N	0001101010
O	0111100001
P	010111010
Q	01111000001111
R	01111000000
S	011110011
T	11111000

U	011110000010
V	00011011100110
W	010111001
X	0111100000111011
Y	00011011111
Z	011110000011100000
[	000110100010
]	000110100001
a	1000
b	111111

c	011000
d	01010
e	110
f	011111
g	011100
h	1010
i	1110
j	0111101011
k	0101111
l	01101

m	011001
n	1011
o	1001
p	0000101
q	00011010110
r	00010
s	00000
t	0100
u	000011
v	1111101

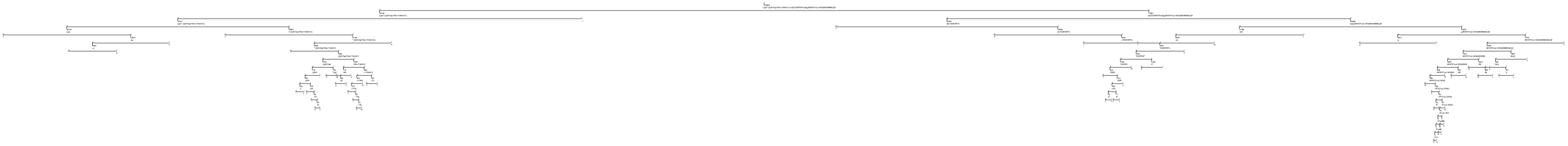
w	000111
x	0111101001
y	010110
z	000110111000

Huffman-Code Tabelle für den Roman Peter Pan mit Vor- und Nachwort vom Projekt [www.gutenberg.org.](http://www.gutenberg.org)

In den folgenden zwei Seiten können Sie den Huffman-Baum mit seinen entstandenen Pfaden genauer untersuchen. Aufgrund der enormen Größe des Baumes wurden die ersten zwei Nodes auf jeweils einer Seite aufgeteilt und es ist starkes Zoomen erforderlich!

Huffman-Baum (Linke Seite vom Startpunkt)

Bitte Zoomen



Huffman-Baum (Rechte Seite vom Startpunkt)

Bitte Zoomen