

# Übungsaufgabe 05 zu Algorithms and Optimizations

14.11.2015

## Gruppenmitglieder:

Tu Tran Thi Ngoc: 537318

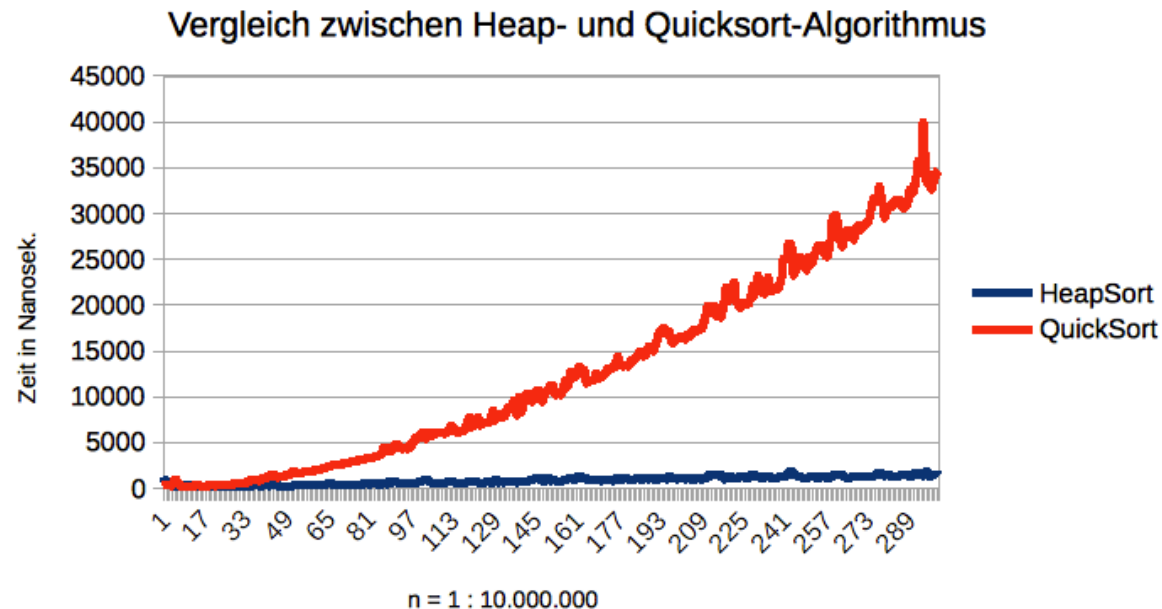
André Vallentin: 527538

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:

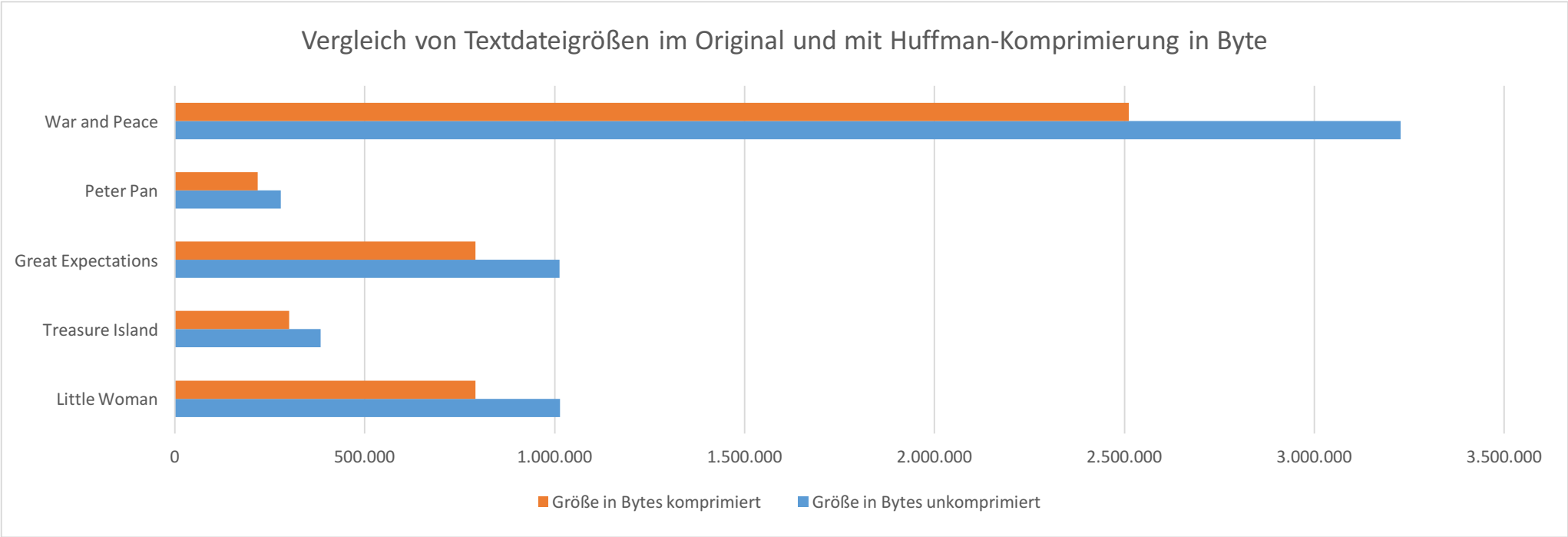
**Schreiben Sie ein Programm, mit dem Sie ein Textfile Huffman-codieren können. Welche Reduktion der Filegröße erreichen Sie damit?**

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	791.606	21,91
Treasure Island	383.673	300.158	21,77
Great Expectations	1.012.626	790.916	21,89
Peter Pan	278.144	217.319	21,87
War and Peace	3.226.620	2.510.942	22,18

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 22 %, worin die meisten Zeichensätze der ASCII-Tabelle im Huffman-Baum enthalten sind.



\n	0100111	*	0011101	5	0010011	A	0001001	K	111111	U	110101		c	101011	m	100001	w	010111
	0100110	,	0011100	6	0010010	B	0001000	L	111110	V	110100		d	101010	n	100000	x	010110
!	0100101	-	0011011	7	0010001	C	0000111	M	111101	W	110011		e	101001	o	011111	y	010101
"	0100100	.	0011010	8	0010000	D	0000110	N	111100	X	110010		f	101000	p	011110	z	010100
#	0100011	/	0011001	9	0001111	E	0000101	O	111011	Y	110001		g	100111	q	011101		
\$	0100010	0	0011000	:	0001110	F	0000100	P	111010	Z	110000		h	100110	r	011100		
%	0100001	1	0010111		0001101	G	0000011	Q	111001	[	101111		i	100101	s	011011		
'	0100000	2	0010110	=	0001100	H	0000010	R	111000	]	101110		j	100100	t	011010		
(	0011111	3	0010101	?	0001011	I	0000001	S	110111	a	101101		k	100011	u	011001		
)	0011110	4	0010100	@	0001010	J	0000000	T	110110	b	101100		l	100010	v	011000		

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