

Hausaufgabe 3

Aaron Sastry

3 mei 2022

Aufgabe 3.1 Radix-Sort

a)

Sortieren Sie die Folge $a = (271, 842, 172, 828, 904, 023, 991, 800, 601, 889)$ mit LSD-RadixSort

	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9
<i>1st 10 buckets</i>	800	271	842	023	904				828	889
		991	172							
		601								

output 1: (800, 271, 991, 601, 842, 172, 023, 904, 828, 889)

	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9
<i>2nd 10 buckets</i>	800		023		842			271	889	991
	601		828					172		
	904									

output 2: (800, 601, 904, 023, 828, 842, 271, 172, 889, 991)

	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9
<i>3rd 10 buckets</i>	023	172	271				601		800	904
									828	991
									842	
									889	

output 3: (023, 172, 271, 601, 800, 828, 842, 889, 904, 991)

Nach 3 Iterationen ist die List final sortiert mit der Reihenfolge:

023, 172, 271, 601, 800, 828, 842, 889, 904, 991

b)

Sortieren Sie die Folge $b = (0.271, 0.842, 0.172, 0.828, 0.904)$ mit MSD-RadixSort.

1st set of buckets

B0	B1	B2	B3	B4	B5	B6	B7	B8	B9
	0.172	0.271						0.842 0.828	0.904

output: 1 = ([0.172], [0.271], [0.842, 0.828], [0.904])

Jetzt werden nur noch die Buckets sortiert die mehr als 1 Element haben.

2nd set of buckets, (the 0.8 bucket)

B0	B1	B2	B3	B4	B5	B6	B7	B8	B9
		0.828		0.842					

output: ([0.172], [0.271], [[0.828], [0.842]], [0.904])

Zu guter Letzt werden jetzt noch die Buckets wieder zu einer fertigen Liste zusammen gesetzt:

$\Rightarrow 0.172, 0.271, 0.828, 0.842, 0.904$

c)

Angenommen, Sie wollen m-stellige Binärzahlen mit MSD-RadixSort sortieren, wobei die Behälter rekursiv wieder mit MSD-RadixSort sortiert werden sollen. Was ist die Laufzeit dieses Sortieralgorithmus für n Binärzahlen, abhängig von n und m? Begründen Sie Ihre Behauptung kurz.

Antwort:

Aus jeder liste entstehen immer 2 buckets auf denen wieder sortiert werden muss. Somit sind in der iteration i etwa : 2^i befüllte Buckets. Da alle Buckets vereinigt wieder genau n Elemente haben ist jede Iteration in

$$T = n$$

Nun gibt es natürlich etwa 2^m Buckets und exakt m operationen da dies die Anzahl der stellen sind in die wir einteilen. Somit erfolgt die Einteilung in die gesamten Buckets in

$$T = m * n$$

Zu guter Letzt fehlt noch die vereinigung der Buckets wieder in eine vollständige List, was wieder genau $T = n$ braucht. Somit ist die finale Laufzeit bei:

$$\begin{aligned}
T = (m + 1) * n &\iff T = m * n + n \\
&\implies \mathcal{O}(m * n)
\end{aligned}$$