

ADP Aufgabe 3

3.A1 ShellSort

1. Bestimmen Sie für $N=12233$ den maximal möglichen Abstand für das ShellSort-Verfahren.
2. Gegeben das unsortierte `int[]`-Array. Nehmen Sie eine 13-Sortierung vor. Zeigen Sie für jedes i den Zustand nach einem Sortierdurchlauf.
3. Welche Überlegung führte zur Einführung von ShellSort.

3.A2 MergeSort

1. Implementieren Sie ein MergeBottomUp Verfahren, das in jedem Durchlauf jeweils 3 sortierte Teilarrays zusammenführt.
2. Berechnen Sie das Laufzeitverhalten für allgemeines N auf Basis der Implementierung.
3. Welche Optimierungen kennen Sie für MergeSort.

3.A3 QuickSort

1. Implementieren Sie QuickSort mit Median von 3 und Median von 7.
2. Vergleichen Sie experimentell die Implementierungen untereinander und mit der Standardimplementierung. Nutzen Sie den Client `SortCompare`.



3.A4 Stabile Sortierverfahren

Zeigen Sie die Ergebnisse eines stabilen Sortierverfahrens, das zuerst nach Datum, dann nach Station, dann nach Ort sortiert.

Ort	Station	Zeit
Hamburg	s1	09:53:12
Kiew	s1	12:28:12
Warschau	s3	10:05:12
Warschau	s2	09:53:12
Kiew	s1	13:00:12
Hamburg	s2	07:46:12
Hamburg	s3	11:45:12
Hamburg	s2	09:42:12
Kiew	s2	12:09:12
Hamburg	s1	09:13:12
Kiew	s3	12:53:12
Oslo	s3	12:39:12
Warschau	s3	12:46:12
Warschau	s2	08:01:12
Oslo	s3	07:45:12
Hamburg	s2	13:12:12
Oslo	s1	10:14:12
Oslo	s1	11:18:12
Kiew	s2	08:44:12
Kiew	s3	06:49:12
Warschau	s1	12:28:12
Oslo	s1	07:06:12
Warschau	s3	10:49:12
Warschau	s1	07:10:12
Warschau	s1	10:36:12
Oslo	s3	10:25:12
Kiew	s3	08:17:12
Kiew	s2	13:40:12
Oslo	s2	10:31:12
Hamburg	s1	06:55:12
Hamburg	s3	12:22:12
Oslo	s2	12:24:12



3.A5 Priority Queues und HeapSort

1. Sie sollen aus einem sehr großen Datensatz, der nicht vollständig in den Speicher passt, die K kleinsten Elemente bestimmen. Implementieren Sie die Lösung. ✓
2. Gegeben die Eingabe S C H L A G L I C H T E R
a. Stellen Sie die Heapordnung nach dem optimierten Verfahren her. ✓
b. Sortieren Sie die ersten vier Elemente.

3.A6 Symboltabellen Rot-Schwarz-Bäume (RB-BST)

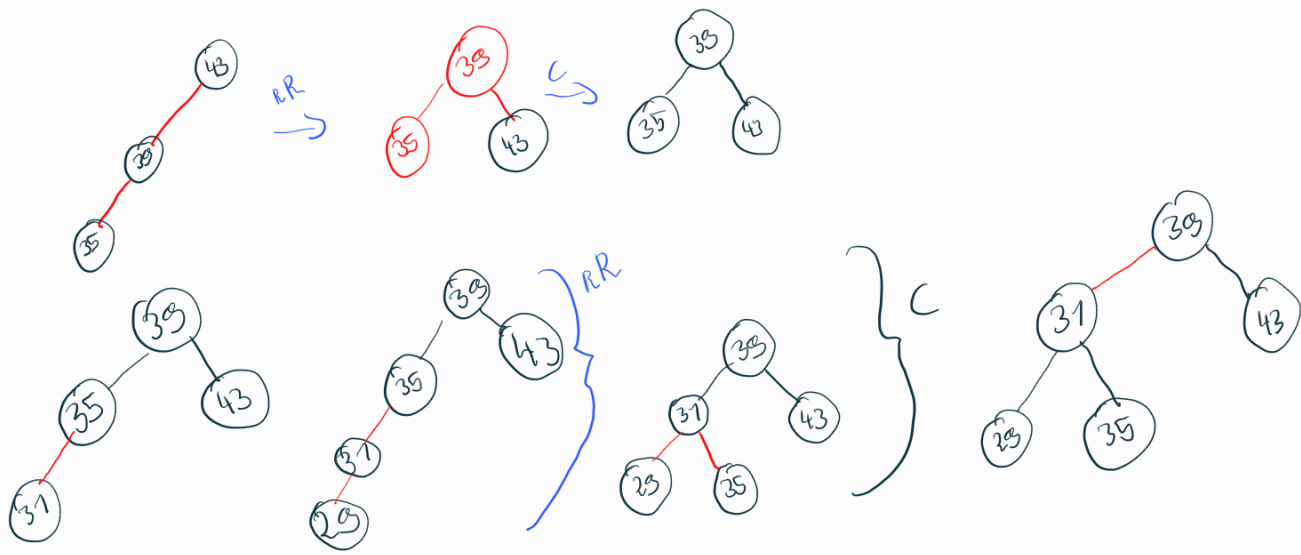
1. Welche zwei Kategorien von Symboltabellen kennen Sie? ✓
2. Leiten Sie für den BinarySearchST das Laufzeitverhalten für den schlechtesten Fall her. ○
3. Konstruieren Sie einen RB-BST für die folgende Eingabe:

Werte/Values
Key/Schlüssel

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
43	39	35	31	29	25	22	21	20	17	14	11	8	5	3	1

4. Verallgemeinern Sie Ihre Beobachtung unter 3.A6.3.
5. Zeigen Sie für den RB-BST die Einzelschritte zur Berechnung von
 - a. `rank(15)`
 - b. `select(24)`
 - c. `keys(1, 24)`.

$\text{Rank}(15) \Rightarrow \text{Lari: Key 15 not found}$
 $\Rightarrow \text{Fabi: } \text{weil Index 15} \rightarrow \text{Key} = 1 \rightarrow \text{keine Kinder}$
 $\text{Select}(24) \Rightarrow \text{gibtes nicht } ???$

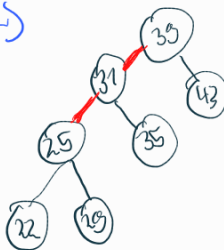
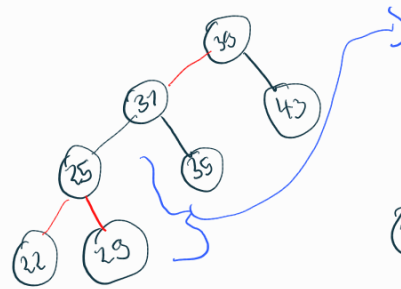
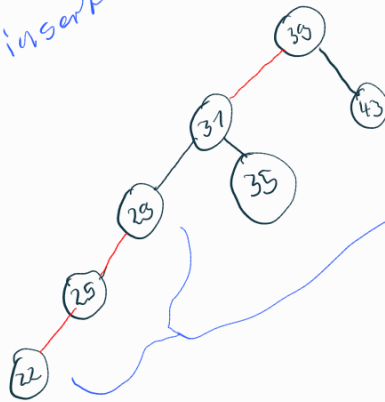
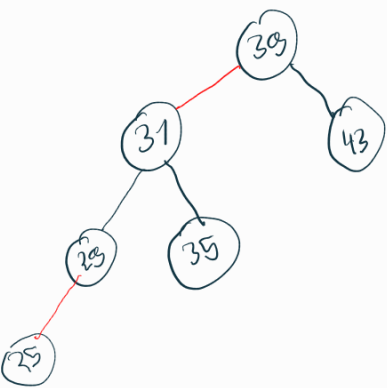


insert 25

insert 22

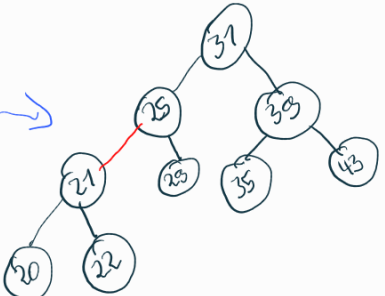
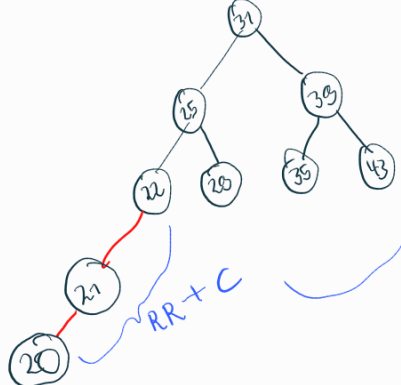
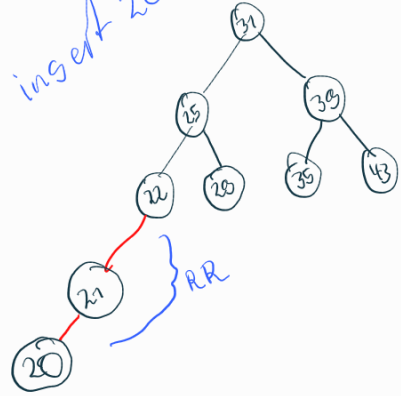
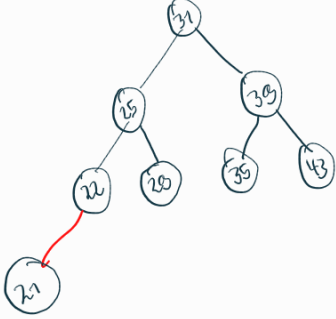
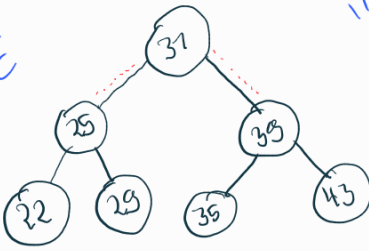
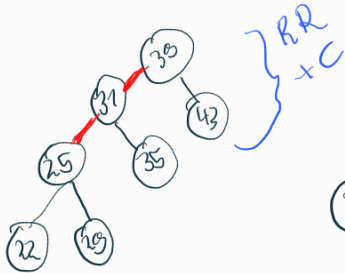
RR

LL



insert 27

insert 20



insert 17

