

TRAVERSIERUNG:

DEPTH-FIRST = STACK BREADTH-PIRST = QUEUE

NODE

⇒ PRÄORDER O(n)

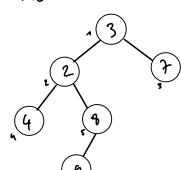
INDROER O(n)

DEPTA-FIRST

> => POSTORDER O(n)

THE:

OUTPUT: 3,2,7,4,8,6



QUEUE: 1. (3) 5. (8)
2. (2,7) 6. (6)
3. (7,4,8) 7. ()

BREADTH - FIRST (LEVEL-DROFFR)
O(n)

SORTS:

INSERTION SORTS

- > STARTE LINKS UNO VERGLEICHE DAS NÄCHSTE ELEMENT
 - MENN DIE REIHENFOLGE PASST, MARKIERE ALS SORTIERT ANSONSTEN SWAPE HIT LINKEN ELEMENT IBIS REIHENFOLGE PASST
- > 0(n2)
- > IN-PLACE UND STABIL

SELECTION SORT:

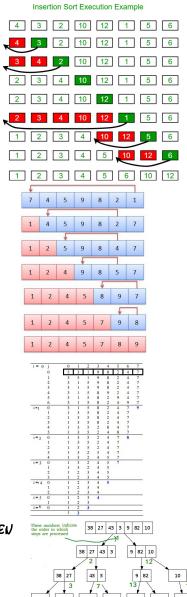
- > STARTE MIT INEL POINTER, P1: MIMIMUM P2: CLERENT ELEMENT
- > TAUSCHE INDEX MIT MINIMUM (VERGLEICHE INDEX MIT ALLEN ELEMENTE
- $> O(n^2)$
- > IN-PLACE UND STANDARDMÄPIG NICHT STABIL

BUBBLE SORT:

- > VERGLEICHE IMMER ZWEI ELEMENTE, WENN DAS ZWEITE ELEMENT ULEINER 157
- > (20)
- > IN-PLACE UND STABIL

MERGE SORTS

- > REKURSIV, DIVIDE & CONQUER
- > QUERST WIRD HALBIERT BIS ALLE ELEMENTE ALLEIN STEHEN, DANN WERDEN IMMER ZWEI SUBARRAYS SORTICAT UND GEMERGED
- > O(nlogn)
- > NOT IN-PLACE UND STABIL



$\textbf{Example:_} \ \, \textbf{The fig. shows steps of heap-sort for list (2\ 3\ 7\ 1\ 8\ 5\ 6)}$ HEAP SORT: 63512**78** > BUILD MAX HEAP > REMOVE ROOT AND SWAP IT WITH THE LAST NODE > CHECK IF IT'S STILL MAX HEAP, IF NOT HEAPIFY > BUILD MAX HEAP O(n) HEAPIFY O(logn) > IN-PLACE AND NOT STABLE initial array 6 4 5 8 2 3 1 9 5 choose pivot for left half; use base case for right 2 4 5 1 3 5 8 9 6 6 4 5 8 2 3 1 9 5 RUICK SORT: 2 4 1 3 5 5 6 8 9 arrange values 2 4 5 8 9 3 1 5 6 りん > RECURSIVE USING PIVOT > CHOOSE PIVOT BY MEDIAN , MOVE ALL ITEMS WHICH 2 1 4 3 5 5 6 8 9 increment 2 4 5 8 9 3 1 5 6 11 ARE SMALLER TO THE LEFT AND THE REST TO THE RIGHT swap values 2 4 5 1 9 3 8 5 6 2 1 4 3 5 5 6 8 9 increment, > WORST CASE: O(n2) 人丿 AVERAGE CASE: () (n log n) increment 2 4 5 1 9 3 8 5 6 move pivot to 2 1 3 4 5 5 6 8 9 final position 11 swap values 2 4 5 1 3 9 8 5 6 2 1 3 4 5 5 6 8 9 remaining pieces 2 4 5 1 3 9 8 5 6 1 2 3 4 5 5 6 8 9 final result REO-BLACK TREE: (BALANCED SEARCHTREE) 2 4 5 1 3 5 8 9 6

SEARCH $O(\log n)$ INSERT $O(\log n)$ REQUIRE ROTATIONS

REMOVE $O(\log n)$

- ! ROOT AND LEAVES ARE ALWAYS BLACK
- ! IF A NODE IS RED , THEN THE CHILDREN ARE RED!

 ALL PATHS FROM A NODE TO ITS LEAVES, HAVE THE SAME NUMBER OF BLACK MODES

 (ROOT EXCLUDED)