algo

Theory & Sort (A, P, r)

Theory & Contine Sorted P-2 and Sorted (3+1) - r

mergésort in Schame

(5, 2, 4, 6, 1, 3, 2, 6)

(5, 2, 4, 6) (1, 3, 2, 6)

(5, 2) (4, 6) (1, 3) (2, 6)

(5) (2) (4) (6) (1) (3) (2) (6)

(2,5) (4, 6) (1, 3) (2, 6)

(2,4,5,6) (1,2,3,6)

(1,2,2,3,4,5,6,6)

Sorted.

marge Sort (5,2,4,6,1,3,2,6) - merge Sort (L) Speit (3,2,96,03,2)6) L if (14 ≥ 2) Isplit (L) SL2 marge Sort (5,4,1,2) marge Sort (2,6,3,6) merge-Sort (L1): > L1 (Sorted L1) Spet (5,4,1,2) Spet (2,6,3,6) merge-Sort (L2); → L2 (Sortad)2) marge Sort (5,1) ms (4,2) ms (2,3) ms (6,6) merge (Li, Li); -> L' (sorted L) Spect (5,1) 5,04 (4,2) 5,04 (2,3) 5,04(6,6) ms(5) ms(1) ms(4) ms(2) ms(2) ms(3) ms(6) m6 Dere return L; Mago(4,2) morge(2,3) more(6,6 -merge-Sort (L) (2,4) [H=&Jor1 ?-> Yes [(2,3) (6,6) morge ((15)(2,4)) Worse (2,3), (6,6)) Sylit (L) (1,2,4,5) (2,3,6,6) merge Sort (4) [Marge Sort (b) merge ((1,2,4,5) (2,3,6,6)) merge(L1, L2) (1,2,2,3,4,5,6,6)

Seudo coda werge (x, Y) E py (x is null), return Y: Lake of (Yie null), return X; Lelse j (1st-ele. 2 x < 1st-ele. 2 Y) Construct (1st-ele.oxX, - olse Construct (1st-ele of), merzo (), merge_sort (x) $\{ (i) (X | i) \}$, $sittam (X; -(1) \Rightarrow (1) \}$ $= \{ (i) (X | i) \}$, $sittam (X; -(1) \Rightarrow (1) \}$. netionmerge (merge sort (9), merge sort (9)) (cdr (split x)) (Car (Split X)) - Split (x) - - (1324) =) ((12)34) Epig (x li mull), construct (()); - (Cons 10 10) felse if (X is 1-ale. list), construct ((1)); (1) -> ((1)) (1324) + ((12) 34) Jeetter of ((13)24) / hok.