

NÅGRA FALL DÅ MAN KAN SORTERA I $O(n \log n)$

1. BARA ETT KONSTANT ANTAL OLIKA ELEMENT SKA SORTERAS.
 $O(n)$ MED RÄKNE SORTERING (COUNTING SORT)
2. ELEMENTEN SOM SKA SORTERAS ÄR TAL SOM ÄR JÄMNT FÖRDELADE I ETT VISST INTERVALL.
 $O(n)$ MED BUCKET SORT
3. ELEMENTEN SOM SKA SORTERAS ÄR STRÄNGAR SOM BESTÅR AV d "SIFFROR". ($v[i] = s_{i,1} s_{i,2} \dots s_{i,d}$)
 $O(nd)$ MED RADIXSORTERING:

FÖR $i \leftarrow d$ DOWNTO 1 DO
SORTERA $v[1..n]$ EFTER SIFFRA i MED
EN STABIL SORTERINGSALGORITM

- Om d är konstant får vi linjär tidskomplexitet.
- Om vi räknar antalet siffror i indata får vi linjär tidskomplexitet $O(N)$ där $N = nd$.

RÄKNE SORTERING

COUNTING SORT ($V[1..n]$, $f: \text{ELEMENT} \rightarrow [1..k]$) =

HJÄLPARRAY FÖR RÄKNING: $C[1..k]$

HJÄLPARRAY FÖR LAGRING AV RESULTATET: $RES[1..n]$

FÖR $i \leftarrow 1$ TO k DO $C[i] \leftarrow 0$

FÖR $j \leftarrow 1$ TO n DO $C[f(V[j])]++$

$sum \leftarrow 0$

FÖR $i \leftarrow k$ DOWNTO 1 DO

$sum \leftarrow sum + C[i]$

$C[i] \leftarrow n - sum + 1$

FÖR $j \leftarrow 1$ TO n DO

$RES[C[f(V[j])]] \leftarrow V[j]$

$C[f(V[j])]++$

RETURN RES

TIDSKOMPLEXITET: $O(k+n)$

EXEMPEL:

$V[1..10] = [7 \ 2 \ 2 \ 4711 \ 2 \ 17 \ 17 \ 4711 \ 2 \ 17]$

$f(2) = 1$, $f(17) = 2$, $f(4711) = 3$

$C[1..3]$ EFTER RÄKNESLINGAN = $[4 \ 4 \ 2]$

$C[1..3]$ EFTER INDEXORÄKNING = $[1 \ 5 \ 9]$

$RES[1..10] = [2 \ 2 \ 2 \ 2 \ 17 \ 17 \ 17 \ 17 \ 4711 \ 4711]$

RADIXSORTERINGSEXEMPEL MED $d=3$

OSORTERAT:	PASS 1:	PASS 2:	PASS 3:
480	480	902	009
973	381	905	381
902		009	
905	902	816	419
532	532	419	480
652	652		
783		532	532
009	973		652
653	783	652	653
419	653	653	
816	905	973	783
381			816
	816	480	
		381	902
	009	783	905
	419		
			973