

Quicksort

Quicksort er en rekursiv algoritme, som bruker "divide and conquer" for sortering.

Den partisjonerer arrayen i 2 deler (left & right), så sorterer disse hver for seg.

Hver gang det skal sorteres, velger en "pivot" hvor verdien høyere enn denne legger seg på dens høyre side, og verdier mindre legges på dens venstre side. I dette eksempelet blir den verdien til "høyre" i gjeldende array/sub-array valgt som pivot. Andre eksempler kan velge denne basert på "Median-of-Three".

Quicksort er en **in-place sorting algoritme**, det vil si at den ikke bruker "hjelpe" arrayer for å sortere.

Quicksort er **ikke stable**, det vil si at keys av samme verdi (eks E og E) vil bytte plass.

Forklaringer

*i itererer opp helt til den ikke lenger er mindre enn *"Pivot"

*j itererer ned helt til den ikke lenger er større enn *"Pivot"

*"Pivot". Elementer mindre skal på dens venstre side. Elementer større skal på dens høyre side, helt til *i og *j krysser

Disse feltene betyr at de ble swappet av den forrige.

Quicksort - kall 1

Quicksort blir kalt med 1 som venstre, 12 som høyre. Siden $12 > 1$ så kjøres resten av koden.

Velger det til høyre som pivot: *N

0	1	2	3	4	5	6	7	8	9	10	11	12
-1	E	A	S	Y	Q	U	E	S	T	I	O	N
			*i							*j		
i >= j? (3 >= 10)? Nei, swap(i, j)												
i)	E < N? A < N? S < N?											
j)	I > N? O > N?											

-1	1	2	3	4	5	6	7	8	9	10	11	12
	E	A	I	Y	Q	U	E	S	T	S	O	N
			*i				*j					
i >= j? (3 >= 7)? Nei, swap(i, j)												
i)	Y < N?											
j)	E > N? S > N? T > N?											

-1	1	2	3	4	5	6	7	8	9	10	11	12
	E	A	I	E	Q	U	Y	S	T	S	O	N
			*j	*i								
I og J krysset, swap(i, pivots indeks)												
i)	Q < N?											
j)	E > N? Q > N? U > N?											

-1 E A I E N U Y S T S O Q Quicksort 1 er ferdig med "partisjonering"

Quicksort kall 1-1

Quicksort blir kalt med 1 som venstre, 4 som høyre (i-1). Siden $4 > 1$ så kjøres resten av koden.

Velger det til høyre som pivot: *E

0	1	2	3	4
-1	E	A	I	E
	*i	*j		
i >= j? Nei, swapp(i, j)				
i)	E < E?			
j)	A > E? I > E?			

-1	1	2	3	4
	A	E	I	E
	*j	*i		
I og J krysset, swap(i, pivots indeks)				
i)	E < E?			
j)	A > E?			

-1 A E I E Quicksort 1-1 er ferdig med "partisjonering"

Quicksort kall 1-1-1

Quicksort blir kalt med 1 som venstre, 1 som høyre (i-1). Siden $1 < 1$ ikke er sant, så kjøres ikke koden.

Quicksort kall 1-1-2

Quicksort blir kalt med 3 som venstre (i+1), 4 som høyre. Siden $4 > 3$ så kjøres resten av koden

Velger det til høyre som pivot: *E

3	4
I	E
*i	*j
i >= j? Nei, swapp(i, j)	
i)	I < E?
j)	E > E?

E	I
*j	*i
I og J krysset, swap(i, pivots indeks)	
I < E?	
E > E?	

E I Quicksort 1-1-2 er ferdig med "partisjonering"

Quicksort kall 1-1-2-1

Quicksort blir kalt med 3 som venstre, 3 som høyre (i-1). Siden $3 > 3$ ikke er sant, så kjøres ikke koden.

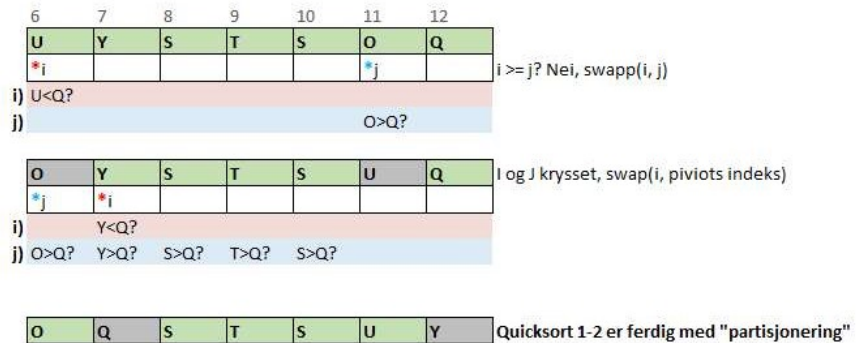
Quicksort kall 1-1-2-2

Quicksort blir kalt med 5 som venstre ($i+1$), 4 som høyre. Siden $4 > 5$ ikke er sant, så kjøres ikke koden.

Quicksort kall 1-2

Quicksort blir kalt med 6 som venstre ($i+1$), 12 som høyre. Siden $12 > 6$ så kjører koden.

Velger det til høyre som pivot: *Q



Quicksort kall 1-2-1

Quicksort blir kalt med 6 som venstre, 6 som høyre ($i-1$). Siden $6 > 6$ ikke er sant, så kjøres ikke koden.

Quicksort kall 1-2-2

Quicksort blir kalt med 7 som venstre ($i+1$), 12 som høyre. Siden $12 > 7$ så kjører koden.

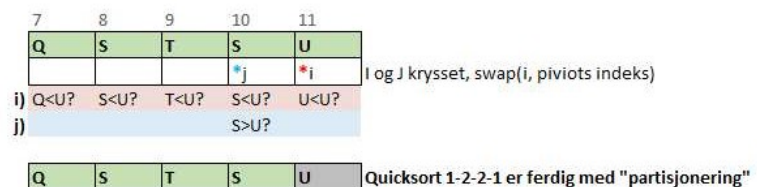
Velger det til høyre som pivot: *Y



Quicksort kall 1-2-2-1

Quicksort blir kalt med 7 som venstre, 11 som høyre. Siden $11 > 7$ så kjører koden

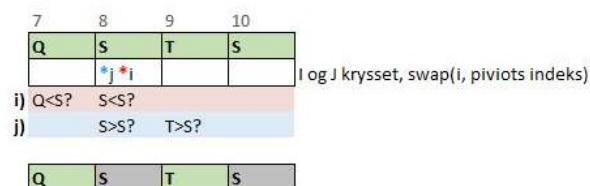
Velger det til høyre som pivot: *U



Quicksort kall 1-2-2-1-1

Quicksort blir kalt med 7 som venstre, 10 ($i-1$) som høyre. Siden $11 > 7$ så kjører koden.

Velger det til høyre som pivot: *S



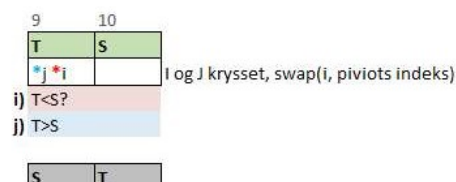
Quicksort kall 1-2-2-1-1-1

Quicksort blir kalt med 7 som venstre, 7 som høyre ($i-1$). Siden $7 > 7$ ikke stemmer, så kjøres ikke koden.

Quicksort kall 1-2-2-1-1-2

Quicksort blir kalt med 9 som venstre ($i+1$), 10 som høyre. Siden $10 > 9$ så kjører koden.

Velger det til høyre som pivot: *S



Quicksort kall 1-2-2-1-1-2-1

Quicksort blir kalt med 9 som venstre, 8 ($i-1$) som høyre. Siden $8 > 9$ ikke stemmer, så kjøres ikke koden.

Quicksort kall 1-2-2-1-1-2-2

Quicksort blir kalt med 10 som høyre ($i+1$), 10 som venstre. Siden $10 > 10$ ikke stemmer, så kjører ikke koden.

Quicksort kall 1-2-2-1-2

Quicksort blir kalt med 12 som venstre ($i+1$), 11 som høyre. Siden $11 > 10$ ikke stemmer, så kjører ikke koden.

Quicksort kall 1-2-2-2

Quicksort blir kalt med 13 som venstre ($i+1$), 12 som høyre. Siden $12 > 13$ ikke stemmer, så kjører ikke koden.

Quicksort er ferdig!

Under er sub-arrayene, lagt sammen vil resultatet være en sortert array.

	0	1	2	3	4	5	6	7	8	9	10	11	12
1)	-1	E	A	I	E	N	U	Y	S	T	S	O	Q
2)	-1	A	E	I	E								
3)				E	I								
4)							O	Q	S	T	S	U	Y
5)								Q	S	T	S	U	Y
6)								Q	S	T	S	U	
7)								Q	S	T	S		
8)										S	T		
	-1	A	E	E	I	N	O	Q	S	S	T	U	Y