

Mergesort

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

Mergesort vil først dele opp arrayen i flere deler, før den sorterer hver sub-array.

Dette fører til at etter hver gang en sub-arrays venstre og høyre side blir sortert, så vil den som kalles mergesort for disse "merge" disse to delene sammen.

Mergesort er en **out-of-place sorting algoritme**, det vil si at den bruker en "hjelp" array for å sortere.

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

Forklaringer

*i - Itererer oppover, sammenlignes med *k
*j - Itererer nedover, sammenlignes med *k
*k - Den indeksen jeg skal sette inn i a[] arrayen.
Markerer b[] (hjelp) arrayen.

Forklaring av "sorteringen"

Bytt den minste inn i *k
Hvis *j bytter, la den gå bakover
Hvis *i bytter, la den gå fremover

Forklaring av opprettelse av b[] array

i går fra "midten" og nedover så lenge den er større enn "l".
Den kopierer a[i-1] til b[i-1] (Mao, en ren kopi av arrayen under "m")
j går fra "midten" og oppover så lenge den er mindre enn "r".
Den kopierer den siste inn i j-1, neste gang den nest siste i j-1 osv.

Mergesort - kall 1

Mergesort blir kalt med 0 som venstre, 11 som høyre. Siden $11 > 0$ så kjøres resten av koden.

Middle = $(11+0)/2 = 5$

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

Mergesort - kall 1-1

Mergesort blir kalt med 0 (l) som venstre, 5 (m) som høyre. Siden $5 > 0$ så kjøres koden.

Middle = $(5+0)/2 = 2$

	0	1	2	3	4	5
a)	E	A	S*	Y	Q	U

Mergesort - kall 1-1-1

Mergesort blir kalt med 0 (l) som venstre, 2 (m) som høyre. Siden $2 > 0$ så kjøres koden.

Middle = $(2+0)/2 = 1$

	0	1	2
a)	E	A*	S

Mergesort - kall 1-1-1-1

Mergesort blir kalt med 0 (l) som venstre, 1 (m) som høyre. Siden $1 > 0$ så kjøres koden.

Middle = $(1+0)/2 = 0$

	0	1
a)	E*	A

Mergesort - kall 1-1-1-1-1

Mergesort blir kalt med 0 (l) som venstre, 0 (m) som høyre. Siden $0 > 0$ ikke stemmer, så kjøres ikke koden.

Mergesort - kall 1-1-1-1-2

Mergesort blir kalt med 1 (m+1) som venstre, 1 (r) som høyre. Siden $1 > 1$ ikke stemmer, så kjøres ikke koden.

Sortering kjøres for mergesort 1-1-1-1

Middle = $(1+0)/2 = 0$ l = 0, r = 1

	0	1
a)	E	A
b)	E	A
b)	E	A

i gkk en gang og er 0 | j gikk en gang og er 1

b)	E	A
	*i *k	*j
a)	A	

$E \leq A$ og $i \leq m$? | Nei og Ja | Sett A(j) i "k", j--

b)	E	A
	*j	*i *k
a)		E

$A \leq E$ og $i \leq m$? | Ja og Nei | Sett E(j) i "k", j--

	0	1
a)	A	E

Sub-array er ferdig sortert

Mergesort - kall 1-1-1-2

Mergesort blir kalt med 2 (m+1) som venstre, 2 (r) som høyre. Siden $2 > 2$ ikke stemmer, så kjøres ikke koden.

Sortering kjøres for mergesort 1-1-1

Middle = $(2+0)/2 = 1$ l = 0, r = 2

	0	1	2
a)	A	E	S

U)	A	E	S
b)			S
b)	A	E	S

i gikk en gang og er 0 | j gikk fra 1-2

b)	A	E	S
	*i *k		*j
a)	A		

A<=S og i<=m? | Ja og Ja | Sett A(i) i "k", i++

b)	A	E	S
		*i *k	*j
a)		E	

E<=S? og i<=m? | Ja og Ja | Sett E(i) i "k", i++

b)	A	E	S
			*i *k *j
a)			S

S<=S? og i<=m? | Ja og Nei | Sett S(j) i "k", j--

a)	A	E	S
----	---	---	---

Sub-array er ferdig sortert

Mergesort - kall 1-1-2

Mergesort blir kalt med 3 (m+1) som venstre, 5 (r) som høyre. Siden 5 > 3 så kjører koden.

Middle = (5+3)/ 2 = 4

	3	4	5
a)	Y	Q*	U

Mergesort - kall 1-1-2-1

Mergesort blir kalt med 3(l) som venstre, 4(m) som høyre. Siden 4 > 3 så kjører koden.

Middle = (3+4)/ 2 = 3

	3	4
a)	Y*	Q

Mergesort - kall 1-1-2-1-1

Mergesort blir kalt med 3(l) som venstre, 3(m) som høyre. Siden 3 > 3 ikke stemmer, så kjører ikke koden.

Mergesort - kall 1-1-2-1-2

Mergesort blir kalt med 4(m+1) som venstre, 4(r) som høyre. Siden 4 < 4 ikke stemmer, så kjører ikke koden.

Sortering kjører for mergesort 1-1-2-1

Middle = (3+4)/ 2 = 3

l = 3, r = 4

	3	4
a)	Y	Q
b)	Y	Q
b)	y	Q

i gikk en gang og er 3 | j gikk en gang og er 4

b)	y	Q
	*i *k	*j
a)	Q	

Y <= Q? og i <= m? | Nei og Ja | Sett Q(j) i "k", j--

b)	y	Q
	*j *i	*k
a)		Y

Y <= Y? og i <= m? | Ja og Ja | Sett Y(i) i "k", i++

a)	Q	Y
----	---	---

Sub-array er ferdig sortert

Mergesort - kall 1-1-2-2

Mergesort blir kalt med 5(m+1) som venstre, 5 (r) som høyre. Siden 5 < 5 ikke stemmer, så kjører ikke koden.

Sortering kjører for mergesort 1-1-2

Middle = (5+3)/ 2 = 4

l = 3, r = 5

	3	4	5
a)	Q	Y	U
b)		Y	U
b)	Q		
b)	Q	Y	U

i gikk fra 4-3 | j gikk en gang og er 5

b)	Q	Y	U
	*i *k		*j
a)	Q		

Q <= U? og i <= m? | Ja og Ja | Sett Q(i) i "k", i++

b)	Q	Y	U
		*i *k	*j
a)		U	

Y <= U? og i <= m? | Nei og Ja | Sett U(j) i "k", j--

b)	Q	Y	U
		*j *i	*k
a)			Y

Y <= Y og i <= m? | Ja og Ja | Sett Y(i) i "k", i++

a)	Q	U	Y
----	---	---	---

Sub-array er ferdig sortert

Sortering kjører for mergesort 1-1

$\text{Middle} = (5+0)/2 = 2$ $l = 0, r = 5$

	0	1	2	3	4	5	
a)	A	E	S	Q	U	Y	
b)			S			Q	$j=2, b[5+2-2] = a[2+1] \mid b[5] = a[3]$
b)		E			U		$j=3, b[5+3-3] = a[3+1] \mid b[4] = a[4]$
b)	A			Y			$j=4, b[5+3-4] = a[4+1] \mid b[3] = a[5]$
b)	A	E	S	Y	U	Q	i gikk fra 2-0 j gikk fra 3-5
b)	A	E	S	Y	U	Q	
a)	*i *k					*j	$A \leq Q? \text{ og } i \leq m? \mid \text{Ja og Ja} \mid \text{Sett } A(i) \text{ i "k", } i++$
a)	A						
b)	A	E	S	Y	U	Q	
a)		*i *k				*j	$E \leq Q? \text{ og } i \leq m? \mid \text{Ja og Ja} \mid \text{Sett } E(i) \text{ i "k", } i++$
a)	E						
b)	A	E	S	Y	U	Q	
a)			*i *k			*j	$S \leq Q? \text{ og } i \leq m? \mid \text{Nei og Ja} \mid \text{Sett 'Q'(j)} \text{ i "k", } j--$
a)			Q				
b)	A	E	S	Y	U	Q	
a)			*i	*k	*j		$S \leq U? \text{ og } i \leq m? \mid \text{Ja og Ja} \mid \text{Sett } S(i) \text{ i "k", } i++$
a)			S				
b)	A	E	S	Y	U	Q	
a)			*i	*k *j			$Y \leq U? \text{ og } i \leq m? \mid \text{Nei og Nei} \mid \text{Sett } U(j) \text{ i "k", } j--$
a)				U			
b)	A	E	S	Y	U	Q	
a)			*j *i		*k		$Y \leq Y \text{ og } i \leq m? \mid \text{Nei og Nei} \mid \text{Sett } Y(j) \text{ i "k", } j--$
a)					Y		
a)	A	E	Q	S	U	Y	Sub-array er ferdig sortert

Mergesort - kall 1-2

Mergesort blir kalt med 6(m+1) som venstre, 11(r) som høyre. Siden $11 > 6$ så kjører koden.

$\text{Middle} = (11+6)/2 = 8$

	6	7	8	9	10	11
a)	E	S	T*	I	O	N

Mergesort - kall 1-2-1

Mergesort blir kalt med 6(l) som venstre, 8(m) som høyre. Siden $8 > 6$ så kjører koden.

$\text{Middle} = (8+6)/2 = 7$

	6	7	8
a)	E	S*	T

Mergesort - kall 1-2-1-1

Mergesort blir kalt med 6(l) som venstre, 7(m) som høyre. Siden $7 > 6$ så kjører koden.

$\text{Middle} = (7+6)/2 = 6$

	6	7
a)	E*	S

Mergesort - kall 1-2-1-1-1

Mergesort blir kalt med 6(l) som venstre, 6(m) som høyre. Siden $6 > 6$ ikke stemmer, så kjører ikke koden.

Mergesort - kall 1-2-1-1-2

Mergesort blir kalt med 7(m+1) som venstre, 7(r) som høyre. Siden $7 > 7$ ikke stemmer, så kjører ikke koden.

Sortering kjører for mergesort 1-2-1-1

$\text{Middle} = (7+6)/2 = 6$ $l = 6, r = 7$

	6	7	
a)	E	S	
b)	E	S	
b)	E	S	i gikk en gang og er 6, j gikk en gang og er 7
b)	E	S	
a)	*i *k	*j	$E \leq S? \text{ og } i \leq m? \mid \text{Ja og Ja} \mid \text{Sett } E(i) \text{ i "k", } i++$
a)	E		
b)	E	S	
a)		*i *k *j	$S \leq S \text{ og } i \leq m? \mid \text{Ja og Nei} \mid \text{Sett } S(j) \text{ i "k", } j--$
a)		S	
a)	E	S	Sub-array er ferdig sortert

Mergesort - kall 1-2-1-2

Mergesort blir kalt med 8(m+1) som venstre, 8(r) som høyre. Siden $8 > 8$ ikke stemmer, så kjører ikke koden.

Sortering kjører for mergesort 1-2-1

Middle = $(8+6)/2 = 7$ $l = 6, r = 8$

	6	7	8	
a)	E	S	T	
b)	E		T	
b)		S		
b)	E	S	T	i gikk fra 7-6 j gikk en gang og er 8
b)	E	S	T	
	*i *k		*j	$E < T$ og $i \leq m$? Ja og Ja Sett E(i) i "k", i++
a)	E			
b)	E	S	T	
		*i *k	*j	$S < T$? og $i \leq m$? Ja og Ja Sett S(i) i "k", i++
a)		S		
b)	E	S	T	
			*i *k *j	$T < T$ og $i \leq m$? Ja og Nei Sett T(j) i "k", j--
a)			T	
a)	E	S	T	Sub-array er ferdig sortert

Mergesort - kall 1-2-2

Mergesort blir kalt med 9(m+1) som venstre, 11(r) som høyre. Siden $11 > 9$ så kjører koden.

Middle = $(11+9)/2 = 10$

	9	10	11
a)	I	O*	N

Mergesort - kall 1-2-2-1-1

Mergesort blir kalt med 9(l) som venstre, 9(m) som høyre. Siden $9 > 9$ ikke stemmer, så kjører ikke koden.

Mergesort - kall 1-2-2-1-2

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

Sortering kjører for mergesort 1-2-2-1

Middle = $(10+9)/2 = 9$ $l = 9, r = 10$

	9	10	
a)	I	O	
b)	I	O	
b)	I	O	i gikk en gang og er 9 j gikk en gang og er 10
b)	I	O	
	*i *k	*j	$I \leq O$? og $i \leq m$? Ja og Ja Sett I(i) i "k", i++
a)	I		
b)	I	O	
		*i *k *j	$O \leq O$? og $i \leq m$? Ja og Nei Sett O(k) i "k", j--
a)		O	
a)	I	O	Sub-array er ferdig sortert

Mergesort - kall 1-2-2-2

Mergesort blir kalt med 11(m+1) som venstre, 11(r) som høyre. Siden $11 > 11$ ikke stemmer, så kjører ikke koden.

Sortering kjører for mergesort 1-2-2

Middle = $(11+9)/2 = 10$ $l = 9, r = 11$

	9	10	11	
a)	I	O	N	
b)	I		N	
b)		O		
b)	I	O	N	i gikk fra 10-9 j gikk en gang og er 11
b)	I	O	N	
	*i *k		*j	$I \leq N$? og $i \leq m$? Ja og Ja Sett I(i) i "k", i++
a)	I			
b)	I	O	N	
		*i *k	*j	$O \leq N$? og $i \leq m$? Nei og Ja Sett N(j) i "k", j--
a)		N		
b)	I	O	N	
		*j *i	*k	$O \leq O$? og $i \leq m$? Ja og Ja Sett O(i) i "k", i++
a)			O	
a)	I	N	O	Sub-array er ferdig sortert

Sortering kjører for mergesort 1-2

Middle = $(11+6)/2 = 8$ $l = 6, r = 11$

6	7	8	9	10	11
---	---	---	---	----	----

Sortering kjører for mergesort 1

		*	i	*	j			*k			U<=T og i<=m? Nei og Ja Sett T(j) i "k", j--
--	--	---	---	---	---	--	--	----	--	--	--

