

Oppgave

Hvor mange måter kan du skrive et positivt heltall n på, som en sum av ett eller flere positive heltall?

For eksempel, for $n = 3$ er svaret 4:

$$\begin{array}{cccc} 3 & = & 2 & + & 1 & = & 1 & + & 2 & = & 1 & + & 1 & + & 1 \\ 1 & & 2 & & 3 & & 4 & & & & & & & & \end{array}$$

Kan skrives som:

$$f(n) = 1 + \sum_{i=1}^{n-1} f(i),$$

Tenk selv	0:30
Jobb sammen	1:30
Observasjoner	
Løsningsforslag	
Refleksjon	1:00

$$1 + f(5) = \sum_{i=1}^4 f(i)$$

5 1 stk. med 5 først

4 + 1 $f(1)$ stk. med 4 først

3 + 2 $f(2)$ stk. med 3 først

2 + 3 $f(3)$ stk. med 2 først

1 + 4 $f(4)$ stk. med 1 først

Oppgave

Hvor mange måter kan du skrive et positivt heltall n på, som en sum av ett eller flere positive heltall?

For eksempel, for $n = 3$ er svaret 4:

$$\begin{array}{cccc} 3 & = & 2 & + & 1 & = & 1 & + & 2 & = & 1 & + & 1 & + & 1 \\ 1 & & 2 & & 3 & & 4 & & & & & & & & \end{array}$$

Kan skrives som:

$$f(n) = 1 + \sum_{i=1}^{n-1} f(i),$$

Tenk selv 0:30

Jobb sammen 1:30

Observasjoner

Løsningsforslag

Refleksjon 1:00

$$1 + f(5) = \sum_{i=1}^4 f(i)$$

5 1 stk. med 5 først

4 + 1 $f(1)$ stk. med 4 først

3 + 2 $f(2)$ stk. med 3 først

2 + 3 $f(3)$ stk. med 2 først

1 + 4 $f(4)$ stk. med 1 først

Oppgave

Hvor mange måter kan du skrive et positivt heltall n på, som en sum av ett eller flere positive heltall?

For eksempel, for $n = 3$ er svaret 4:

$$\begin{array}{cccc} 3 & = & 2 & + & 1 & = & 1 & + & 2 & = & 1 & + & 1 & + & 1 \\ 1 & & 2 & & 3 & & 4 & & & & & & & & \end{array}$$

Kan skrives som:

$$f(n) = 1 + \sum_{i=1}^{n-1} f(i),$$

Tenk selv 0:30

Jobb sammen 1:30

Observasjoner

Løsningsforslag

Refleksjon 1:00

$$1 + f(5) = \sum_{i=1}^4 f(i)$$

5 1 stk. med 5 først

4 + 1 $f(1)$ stk. med 4 først

3 + 2 $f(2)$ stk. med 3 først

2 + 3 $f(3)$ stk. med 2 først

1 + 4 $f(4)$ stk. med 1 først

Oppgave

Hvor mange måter kan du skrive et positivt heltall n på, som en sum av ett eller flere positive heltall?

For eksempel, for $n = 3$ er svaret 4:

$$\begin{array}{cccc} 3 & = & 2 & + & 1 & = & 1 & + & 2 & = & 1 & + & 1 & + & 1 \\ 1 & & 2 & & 3 & & 4 & & & & & & & & \end{array}$$

Kan skrives som:

$$f(n) = 1 + \sum_{i=1}^{n-1} f(i),$$

Tenk selv 0:30

Jobb sammen 1:30

Observasjoner

Løsningsforslag

Refleksjon 1:00

$$1 + f(5) = \sum_{i=1}^4 f(i)$$

5 1 stk. med 5 først

4 + 1 $f(1)$ stk. med 4 først

3 + 2 $f(2)$ stk. med 3 først

2 + 3 $f(3)$ stk. med 2 først

1 + 4 $f(4)$ stk. med 1 først

Oppgave

Hvor mange måter kan du skrive et positivt heltall n på, som en sum av ett eller flere positive heltall?

For eksempel, for $n = 3$ er svaret 4:

$$\begin{array}{cccc} 3 & = & 2 & + & 1 & = & 1 & + & 2 & = & 1 & + & 1 & + & 1 \\ 1 & & 2 & & 3 & & 4 & & & & & & & & \end{array}$$

Kan skrives som:

$$f(n) = 1 + \sum_{i=1}^{n-1} f(i),$$

Tenk selv 0:30

Jobb sammen 1:30

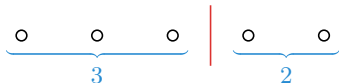
Observasjoner

Løsningsforslag

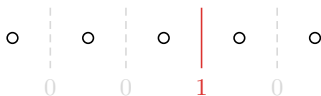
Refleksjon 1:00

Løsningsskisse

Tenk gruppering av n elementer:



Vi har $n - 1$ mulige splittpunkter:



Altså 2^{n-1} muligheter.

$$2^{n-1} = 1 + \sum_{i=1}^{n-1} 2^{i-1}$$

Oppgave

Hvor mange måter kan du skrive et positivt heltall n på, som en sum av ett eller flere positive heltall?

For eksempel, for $n = 3$ er svaret 4:

$$\begin{array}{cccc} 3 & = & 2 & + & 1 & = & 1 & + & 2 & = & 1 & + & 1 & + & 1 \\ 1 & & 2 & & 3 & & 4 & & & & & & & & \end{array}$$

Kan skrives som:

$$f(n) = 1 + \sum_{i=1}^{n-1} f(i),$$

Tenk selv 0:30

Jobb sammen 1:30

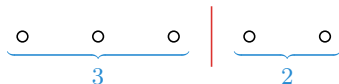
Observasjoner

Løsningsforslag

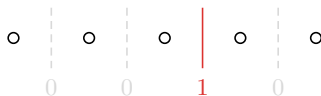
Refleksjon 1:00

Løsningsskisse

Tenk gruppering av n elementer:



Vi har $n - 1$ mulige splittpunkter:



Altså 2^{n-1} muligheter.

$$2^{n-1} = 1 + \sum_{i=1}^{n-1} 2^{i-1}$$

Hva tenkte og gjorde du? Hvorfor? Hva fungerte? Glemte du noe? Hva skjønner du nå? Hvilke nye sammenhenger ser du? Hva skjønner du fortsatt ikke? Hva vil du huske på eller gjøre annerledes senere?

Oppgave

Å avgjøre om en tabell er uten duplikater må beviselig ha kjøretid $\Omega(n \lg n)$ i verste tilfelle.

Hva sier dette om sortering?

Bonus: Tenk på både øvre og nedre grenser i begge retninger.

Ikke nødvendigvis heltall! Verdiene kan kun sammenlignes ($a_i \leq a_j$), ikke brukes som indekser eller hashnøkler.

Tenk selv	0:30
Jobb sammen	1:30
Observasjoner	
Løsningsforslag	
Refleksjon	1:00

$$T_W(n) =$$

$$\Omega(n \lg n)$$

7	3	2	9	3	4	5	1
---	---	---	---	---	---	---	---

Oppgave

Å avgjøre om en tabell er uten duplikater må beviselig ha kjøretid $\Omega(n \lg n)$ i verste tilfelle.

Hva sier dette om sortering?

Bonus: Tenk på både øvre og nedre grenser i begge retninger.

Ikke nødvendigvis heltall! Verdiene kan kun sammenlignes ($a_i \leq a_j$), ikke brukes som indekser eller hashnøkler.

Tenk selv	0:30
-----------	------

Jobb sammen	1:30
-------------	------

Observasjoner

Løsningsforslag

Refleksjon	1:00
------------	------

$$T_W(n) =$$

$$\Omega(n \lg n)$$

7	3	2	9	3	4	5	1
---	---	---	---	---	---	---	---

Oppgave

Å avgjøre om en tabell er uten duplikater må beviselig ha kjøretid $\Omega(n \lg n)$ i verste tilfelle.

Hva sier dette om sortering?

Bonus: Tenk på både øvre og nedre grenser i begge retninger.

Ikke nødvendigvis heltall! Verdiene kan kun sammenlignes ($a_i \leq a_j$), ikke brukes som indekser eller hashnøkler.

Tenk selv 0:30

Jobb sammen 1:30

Observasjoner

Løsningsforslag

Refleksjon 1:00

$$T_W(n) =$$

$$\Omega(n \lg n)$$

7	3	2	9	3	4	5	1
---	---	---	---	---	---	---	---

Oppgave

Å avgjøre om en tabell er uten duplikater må beviselig ha kjøretid $\Omega(n \lg n)$ i verste tilfelle.

Hva sier dette om sortering?

Bonus: Tenk på både øvre og nedre grenser i begge retninger.

Ikke nødvendigvis heltall! Verdiene kan kun sammenlignes ($a_i \leq a_j$), ikke brukes som indekser eller hashnøkler.

Tenk selv 0:30

Jobb sammen 1:30

Observasjoner

Løsningsforslag

Refleksjon 1:00

$$T_W(n) =$$

$$\Omega(n \lg n)$$

7	3	2	9	3	4	5	1
---	---	---	---	---	---	---	---

Oppgave

Å avgjøre om en tabell er uten duplikater må beviselig ha kjøretid $\Omega(n \lg n)$ i verste tilfelle.

Hva sier dette om sortering?

Bonus: Tenk på både øvre og nedre grenser i begge retninger.

Ikke nødvendigvis heltall! Verdiene kan kun sammenlignes ($a_i \leq a_j$), ikke brukes som indekser eller hashnøkler.

Tenk selv 0:30

Jobb sammen 1:30

Observasjoner

Løsningsforslag

Refleksjon 1:00

Løsningsskisse

7	3	2	9	3	4	5	1
---	---	---	---	---	---	---	---

Sortering: X

1	2	3	3	4	5	7	9
---	---	---	---	---	---	---	---

Finn like naboer: $O(n)$

1	2	3	3	4	5	7	9
---	---	---	---	---	---	---	---

$$X + O(n) = \Omega(n \lg n)$$

$$X = \Omega(n \lg n)$$

Øvre grenser virker i motsatt retning.

Oppgave

Å avgjøre om en tabell er uten duplikater må beviselig ha kjøretid $\Omega(n \lg n)$ i verste tilfelle.

Hva sier dette om sortering?

Bonus: Tenk på både øvre og nedre grenser i begge retninger.

Ikke nødvendigvis heltall! Verdiene kan kun sammenlignes ($a_i \leq a_j$), ikke brukes som indekser eller hashnøkler.

Tenk selv 0:30

Jobb sammen 1:30

Observasjoner

Løsningsforslag

Refleksjon 1:00

Løsningsskisse

7	3	2	9	3	4	5	1
---	---	---	---	---	---	---	---

Sortering: X

1	2	3	3	4	5	7	9
---	---	---	---	---	---	---	---

Finn like naboer: $O(n)$

1	2	3	3	4	5	7	9
---	---	---	---	---	---	---	---

$$X + O(n) = \Omega(n \lg n)$$

$$X = \Omega(n \lg n)$$

Øvre grenser virker i motsatt retning.

Hva tenkte og gjorde du? Hvorfor? Hva fungerte? Glemte du noe? Hva skjønner du nå? Hvilke nye sammenhenger ser du? Hva skjønner du fortsatt ikke? Hva vil du huske på eller gjøre annerledes senere?