Øvingsforelesning 2

TDT4120 - Algoritmer og datastrukturer

Øving 1

Logaritmer og modulo

Oppgave 2: Hvilke utsagn om logaritmer stemmer?

Logaritmer og modulo

Oppgave 2: Hvilke utsagn om logaritmer stemmer?

$$\log_a a^n = n$$

Logaritmer og modulo

Oppgave 2: Hvilke utsagn om logaritmer stemmer?

$$\log_a a^n = n$$

Oppgave 3: Hva betyr $a \bmod n = 1$?

Logaritmer og modulo

Oppgave 2: Hvilke utsagn om logaritmer stemmer?

$$\log_a a^n = n$$

Oppgave 3: Hva betyr $a \bmod n = 1$?

$$a = c \cdot n + 1$$

Kjøretidsklasser

Konstant	$\Theta(1)$
Eksponentiell	$\Theta(2^n)$
Kvadratisk	$\Theta(n^2)$
Linearitmisk	$\Theta(n \lg n)$
Faktoriell	$\Theta(n!)$
Kubisk	$\Theta(n^3)$
Logaritmisk	$\Theta(\lg n)$
Lineær	$\Theta(n)$

Kjøretidsklasser

Konstant	$\Theta(1)$
Eksponentiell	$\Theta(2^n)$
Kvadratisk	$\Theta(n^2)$
Linearitmisk	$\Theta(n \lg n)$
Faktoriell	$\Theta(n!)$
Kubisk	$\Theta(n^3)$
Logaritmisk	$\Theta(\lg n)$
Lineær	$\Theta(n)$

$\longrightarrow$

Konstant	$\Theta(1)$
Logaritmisk	$\Theta(\lg n)$
Lineær	$\Theta(n)$
Linearitmisk	$\Theta(n \lg n)$
Kvadratisk	$\Theta(n^2)$
Kubisk	$\Theta(n^3)$
Eksponentiell	$\Theta(2^n)$
Faktoriell	$\Theta(n!)$

Merk: I lysarket brukt i videoen var det en feil her, hvor «Linearitmisk» var plassert feil.

Asymptotisk notasjon

Ønsker å finne sammenhengen mellom veksten til to funksjoner $f(n)$ og $g(n)$.

Oppgave 5: $f(n) = \lg(n^{\lg 5})$ og $g(n) = \lg(5^{\lg n})$

Asymptotisk notasjon

Ønsker å finne sammenhengen mellom veksten til to funksjoner $f(n)$ og $g(n)$.

Oppgave 5: $f(n) = \lg(n^{\lg 5})$ og $g(n) = \lg(5^{\lg n})$

$$f(n) = \lg(n^{\lg 5}) = \lg 5 \lg n = \lg(5^{\lg n}) = g(n)$$

Asymptotisk notasjon

Ønsker å finne sammenhengen mellom veksten til to funksjoner $f(n)$ og $g(n)$.

Oppgave 5: $f(n) = \lg(n^{\lg 5})$ og $g(n) = \lg(5^{\lg n})$

$$f(n) = \lg(n^{\lg 5}) = \lg 5 \lg n = \lg(5^{\lg n}) = g(n)$$

$$f(n) = \Omega(g(n)), f(n) = \mathrm{O}(g(n)), f(n) = \Theta(g(n))$$

Asymptotisk notasjon

Ønsker å finne sammenhengen mellom veksten til to funksjoner $f(n)$ og $g(n)$.

Oppgave 5: $f(n) = \lg(n^{\lg 5})$ og $g(n) = \lg(5^{\lg n})$

$$f(n) = \lg(n^{\lg 5}) = \lg 5 \lg n = \lg(5^{\lg n}) = g(n)$$

$$f(n) = \Omega(g(n)), f(n) = \mathrm{O}(g(n)), f(n) = \Theta(g(n))$$

Oppgave 6: $f(n) = n^2 \lg n + n + n \lg n$ og $g(n) = n^2$

Asymptotisk notasjon

Ønsker å finne sammenhengen mellom veksten til to funksjoner $f(n)$ og $g(n)$.

Oppgave 5: $f(n) = \lg(n^{\lg 5})$ og $g(n) = \lg(5^{\lg n})$

$$f(n) = \lg(n^{\lg 5}) = \lg 5 \lg n = \lg(5^{\lg n}) = g(n)$$

$$f(n) = \Omega(g(n)), f(n) = \mathrm{O}(g(n)), f(n) = \Theta(g(n))$$

Oppgave 6: $f(n) = n^2 \lg n + n + n \lg n$ og $g(n) = n^2$

$$\log_x n > c \quad \text{når } n > x^c$$

Asymptotisk notasjon

Ønsker å finne sammenhengen mellom veksten til to funksjoner $f(n)$ og $g(n)$.

Oppgave 5: $f(n) = \lg(n^{\lg 5})$ og $g(n) = \lg(5^{\lg n})$

$$f(n) = \lg(n^{\lg 5}) = \lg 5 \lg n = \lg(5^{\lg n}) = g(n)$$

$$f(n) = \Omega(g(n)), f(n) = \mathrm{O}(g(n)), f(n) = \Theta(g(n))$$

Oppgave 6: $f(n) = n^2 \lg n + n + n \lg n$ og $g(n) = n^2$

$$\log_x n > c \quad \text{når } n > x^c$$

$$f(n) > c \cdot g(n) \text{ for større verdier av n}$$

Asymptotisk notasjon

Ønsker å finne sammenhengen mellom veksten til to funksjoner $f(n)$ og $g(n)$.

Oppgave 5: $f(n) = \lg(n^{\lg 5})$ og $g(n) = \lg(5^{\lg n})$

$$f(n) = \lg(n^{\lg 5}) = \lg 5 \lg n = \lg(5^{\lg n}) = g(n)$$

$$f(n) = \Omega(g(n)), f(n) = \mathrm{O}(g(n)), f(n) = \Theta(g(n))$$

Oppgave 6: $f(n) = n^2 \lg n + n + n \lg n$ og $g(n) = n^2$

$$\log_x n > c \quad \text{når } n > x^c$$

$$f(n) > c \cdot g(n) \text{ for større verdier av n}$$

$$f(n) = \omega(g(n)), g(n) = \mathrm{O}(f(n))$$

Forenkling av asymptotisk notasjon

Ønsker å forenkle uttrykk med asymptotisk notasjon uten tap av presisjon.

Forenkling av asymptotisk notasjon

Ønsker å forenkle uttrykk med asymptotisk notasjon uten tap av presisjon.

Oppgave 7: Forenkle $\mathrm{O}(n^2) + \Theta(n^3)$

Forenkling av asymptotisk notasjon

Ønsker å forenkle uttrykk med asymptotisk notasjon uten tap av presisjon.

Oppgave 7: Forenkle $\mathrm{O}(n^2) + \Theta(n^3)$

$$\mathrm{O}(n^2) + \Theta(n^3) = \Theta(n^3)$$

Forenkling av asymptotisk notasjon

Ønsker å forenkle uttrykk med asymptotisk notasjon uten tap av presisjon.

Oppgave 7: Forenkle $\mathrm{O}(n^2) + \Theta(n^3)$

$$\mathrm{O}(n^2) + \Theta(n^3) = \Theta(n^3)$$

Oppgave 8: Forenkle $\Omega(n^2) + \Theta(n^3)$

Forenkling av asymptotisk notasjon

Ønsker å forenkle uttrykk med asymptotisk notasjon uten tap av presisjon.

Oppgave 7: Forenkle $\mathrm{O}(n^2) + \Theta(n^3)$

$$\mathrm{O}(n^2) + \Theta(n^3) = \Theta(n^3)$$

Oppgave 8: Forenkle $\Omega(n^2) + \Theta(n^3)$

$$\Omega(n^2) + \Theta(n^3) = \Omega(n^3)$$

Ekvivalens av asymptotisk notasjon

Oppgave 9: Ønsker å finne uttrykk som passer på høyresiden av $\Theta(n^2) + \mathrm{O}(n^4) + \Omega(\lg n) = \underline{\hspace{3cm}}$.

Ekvivalens av asymptotisk notasjon

Oppgave 9: Ønsker å finne uttrykk som passer på høyresiden av $\Theta(n^2) + \mathrm{O}(n^4) + \Omega(\lg n) = \underline{\hspace{3cm}}$.

Ekvivalens er asymmetrisk og høyresiden kan ikke være mer presis enn venstresiden.

Ekvivalens av asymptotisk notasjon

Oppgave 9: Ønsker å finne uttrykk som passer på høyresiden av $\Theta(n^2) + \mathrm{O}(n^4) + \Omega(\lg n) =$ ____________.

Ekvivalens er asymmetrisk og høyresiden kan ikke være mer presis enn venstresiden.

Eksempler

Ekvivalens av asymptotisk notasjon

Oppgave 9: Ønsker å finne uttrykk som passer på høyresiden av $\Theta(n^2) + \mathrm{O}(n^4) + \Omega(\lg n) = \underline{\hspace{3cm}}$.

Ekvivalens er asymmetrisk og høyresiden kan ikke være mer presis enn venstresiden.

Eksempler

$$\Theta(n^2) + \mathrm{O}(n^4) + \Omega(\lg n) = \Omega(n^2) + \mathrm{O}(n^4)$$

Ekvivalens av asymptotisk notasjon

Oppgave 9: Ønsker å finne uttrykk som passer på høyresiden av $\Theta(n^2) + \mathrm{O}(n^4) + \Omega(\lg n) = \underline{\hspace{3cm}}$.

Ekvivalens er asymmetrisk og høyresiden kan ikke være mer presis enn venstresiden.

Eksempler

$$\Theta(n^2) + \mathrm{O}(n^4) + \Omega(\lg n) = \Omega(n^2) + \mathrm{O}(n^4)$$

$$\Theta(n^2) + \mathrm{O}(n^4) + \Omega(\lg n) \neq \Theta(n^2)$$

Forskjellige typer kjøretid

Oppgave 10: Sammenhengen mellom den gjennomsnittlige kjøretiden og den generelle.

Forskjellige typer kjøretid

Oppgave 10: Sammenhengen mellom den gjennomsnittlige kjøretiden og den generelle.

Den gjennomsnittlige ligger innenfor grensene til den generelle.

Forskjellige typer kjøretid

Oppgave 10: Sammenhengen mellom den gjennomsnittlige kjøretiden og den generelle.

Den gjennomsnittlige ligger innenfor grensene til den generelle.

$$\mathrm{O}(\Omega(g(n))), \Omega(\mathrm{O}(g(n)))$$

Forskjellige typer kjøretid

Oppgave 10: Sammenhengen mellom den gjennomsnittlige kjøretiden og den generelle.

Den gjennomsnittlige ligger innenfor grensene til den generelle.

$$\mathrm{O}(\Omega(g(n))), \Omega(\mathrm{O}(g(n)))$$

Oppgave 11: Kjøretiden til en algoritme er beskrevet av $\Omega(n) + \Theta(n^2 \lg n) + \mathrm{O}(n^3)$, hva vet vi om kjøretiden i beste og verste tilfellet?

Forskjellige typer kjøretid

Oppgave 10: Sammenhengen mellom den gjennomsnittlige kjøretiden og den generelle.

Den gjennomsnittlige ligger innenfor grensene til den generelle.

$$\mathrm{O}(\Omega(g(n))), \Omega(\mathrm{O}(g(n)))$$

Oppgave 11: Kjøretiden til en algoritme er beskrevet av $\Omega(n) + \Theta(n^2 \lg n) + \mathrm{O}(n^3)$, hva vet vi om kjøretiden i beste og verste tilfellet?

$$\Omega(n) + \Theta(n^2 \lg n) + \mathrm{O}(n^3) = \Omega(n^2 \lg n)$$

Forskjellige typer kjøretid

Oppgave 10: Sammenhengen mellom den gjennomsnittlige kjøretiden og den generelle.

Den gjennomsnittlige ligger innenfor grensene til den generelle.

$$\mathrm{O}(\Omega(g(n))), \Omega(\mathrm{O}(g(n)))$$

Oppgave 11: Kjøretiden til en algoritme er beskrevet av $\Omega(n) + \Theta(n^2 \lg n) + \mathrm{O}(n^3)$, hva vet vi om kjøretiden i beste og verste tilfellet?

$$\Omega(n) + \Theta(n^2 \lg n) + \mathrm{O}(n^3) = \Omega(n^2 \lg n)$$

Kjenner kun en nedre grense.

INSERTION-SORT

Oppgave 12: Hva stemmer om kjøretiden til INSERTION-SORT?

Oppgave 12: Hva stemmer om kjøretiden til INSERTION-SORT?

Verste tilfelle kjøretid $\Theta(n^2)$
Beste tilfelle kjøretid $\Theta(n)$

Nim

Trekker 1, 2, 3, 4, 5, 6 eller 7 fyrstikker på tur, taper ved å trekke siste fyrstikk.

Nim

Trekker 1, 2, 3, 4, 5, 6 eller 7 fyrstikker på tur, taper ved å trekke siste fyrstikk.

Når og hvordan kan vi garantere å vinne?

Nim

Trekker 1, 2, 3, 4, 5, 6 eller 7 fyrstikker på tur, taper ved å trekke siste fyrstikk.

Når og hvordan kan vi garantere å vinne?

Antall	**Trekke**	**Igjen**	**Vinne**
1	1	0	Nei
2	1	1	Ja
3	2	1	Ja
4	3	1	Ja
5	4	1	Ja
6	5	1	Ja
7	6	1	Ja
8	7	1	Ja
9	?	?	?

Nim

Trekker 1, 2, 3, 4, 5, 6 eller 7 fyrstikker på tur, taper ved å trekke siste fyrstikk.

Når og hvordan kan vi garantere å vinne?

Antall	**Trekke**	**Igjen**	**Vinne**
9	1	8	Nei
9	2	7	Nei
9	3	6	Nei
9	4	5	Nei
9	5	4	Nei
9	6	3	Nei
9	7	2	Nei

Umulig å garantert vinne når det er 9 fyrstikker igjen.

Nim

Trekker 1, 2, 3, 4, 5, 6 eller 7 fyrstikker på tur, taper ved å trekke siste fyrstikk.

Når og hvordan kan vi garantere å vinne?

Antall	**Trekke**	**Igjen**	**Vinne**
10	1	9	Ja
11	2	9	Ja
12	3	9	Ja
13	4	9	Ja
14	5	9	Ja
15	6	9	Ja
16	7	9	Ja

Kan garantert vinne når det er mellom 10 og 16 fyrstikker igjen.

Nim

Trekker 1, 2, 3, 4, 5, 6 eller 7 fyrstikker på tur, taper ved å trekke siste fyrstikk.

Når og hvordan kan vi garantere å vinne?

Kan vinne når $n \neq c \cdot 8 + 1 \Rightarrow n \bmod 8 \neq 1$.

Nim

Trekker 1, 2, 3, 4, 5, 6 eller 7 fyrstikker på tur, taper ved å trekke siste fyrstikk.

Når og hvordan kan vi garantere å vinne?

Kan vinne når $n \neq c \cdot 8 + 1 \Rightarrow n \bmod 8 \neq 1$.

Hvordan beviser vi dette?

Nim

Trekker 1, 2, 3, 4, 5, 6 eller 7 fyrstikker på tur, taper ved å trekke siste fyrstikk.

Når og hvordan kan vi garantere å vinne?

Kan vinne når $n \neq c \cdot 8 + 1 \Rightarrow n \bmod 8 \neq 1$.

Hvordan beviser vi dette?

Allerede bevist for $n < 17$. Kan anta at det gjelder for alle $n < k$ når $17 \leqslant k$ (induksjon). Må vise at det gjelder for $n = k$.

Nim

Trekker 1, 2, 3, 4, 5, 6 eller 7 fyrstikker på tur, taper ved å trekke siste fyrstikk.

Når og hvordan kan vi garantere å vinne?

Kan vinne når $n \neq c \cdot 8 + 1 \Rightarrow n \bmod 8 \neq 1$.

Hvordan beviser vi dette?

Allerede bevist for $n < 17$. Kan anta at det gjelder for alle $n < k$ når $17 \leqslant k$ (induksjon). Må vise at det gjelder for $n = k$.

Hvis $k \bmod 8 = 1$ får vi uansett hvor mange fyrstikker vi trekker, i, at $(k - i) \bmod 8 \neq 1$.

Nim

Trekker 1, 2, 3, 4, 5, 6 eller 7 fyrstikker på tur, taper ved å trekke siste fyrstikk.

Når og hvordan kan vi garantere å vinne?

Kan vinne når $n \neq c \cdot 8 + 1 \Rightarrow n \bmod 8 \neq 1$.

Hvordan beviser vi dette?

Allerede bevist for $n < 17$. Kan anta at det gjelder for alle $n < k$ når $17 \leqslant k$ (induksjon). Må vise at det gjelder for $n = k$.

Hvis $k \bmod 8 = 1$ får vi uansett hvor mange fyrstikker vi trekker, i, at $(k - i) \bmod 8 \neq 1$.

Hvis $k \bmod 8 \neq 1$ kan vi trekke $(k - 1) \bmod 8$ fyrstikker, og ende opp med $(k - i) \bmod 8 = 1$.

K-Largest

```
K-LARGEST(A, k)
let B[1..k] be a new array
for i = 1 to k
    B[i] = 0
for i = 1 to A.length
    if A[i] > B[1]
        B[1] = A[i]
        j = 2
        while j ⩽ B.length and B[j − 1] > B[j]
            exchange B[j − 1] with B[j]
            j = j + 1
sum = 0
for i = 1 to k
    sum = sum + B[i]
return sum
```

K-Largest

```
K-Largest(A, k)
let B[1..k] be a new array
for i = 1 to k
    B[i] = 0
for i = 1 to A.length
    if A[i] > B[1]
        B[1] = A[i]
        j = 2
        while j ⩽ B.length and B[j − 1] > B[j]
            exchange B[j − 1] with B[j]
            j = j + 1
sum = 0
for i = 1 to k
    sum = sum + B[i]
return sum
```

Beste tilfelle $\Theta(n + k^2)$

K-Largest

```
K-LARGEST(A, k)
let B[1..k] be a new array
for i = 1 to k
    B[i] = 0
for i = 1 to A.length
    if A[i] > B[1]
        B[1] = A[i]
        j = 2
        while j ⩽ B.length and B[j − 1] > B[j]
            exchange B[j − 1] with B[j]
            j = j + 1
sum = 0
for i = 1 to k
    sum = sum + B[i]
return sum
```

Beste tilfelle $\Theta(n + k^2)$, verste tilfelle $\Theta(nk)$

K-Largest

Hvordan beviser vi at K-Largest stemmer?

Hvordan beviser vi at K-LARGEST **stemmer?**

Anvender induksjon

K-Largest

Hvordan beviser vi at K-Largest **stemmer?**

Anvender induksjon

Løkkeinvariant: Etter iterasjon *i* gjelder følgende:

K-Largest

Hvordan beviser vi at K-Largest **stemmer?**

Anvender induksjon

Løkkeinvariant: Etter iterasjon i gjelder følgende:

- B består av de $\min(k, i)$ største tallene i $\mathrm{A}[1..i]$, samt $\max(k - i, 0)$ nuller.

K-LARGEST

Hvordan beviser vi at K-LARGEST **stemmer?**

Anvender induksjon

Løkkeinvariant: Etter iterasjon i gjelder følgende:

- B består av de $\min(k, i)$ største tallene i $\mathrm{A}[1..i]$, samt $\max(k - i, 0)$ nuller.
- B er sortert i stigende rekkefølge.

K-Largest

Hvilke alternativer har vi til K-Largest?

K-Largest

Hvilke alternativer har vi til K-Largest?

Vi kan sortere A med en sorteringsalgoritme. Insertion-Sort har

- $\Theta(n)$ i beste tilfelle
- $\Theta(n^2)$ i gjennomsnitt
- $\Theta(n^2)$ i verste tilfelle

K-Largest

Hvilke alternativer har vi til K-Largest?

Vi kan sortere A med en sorteringsalgoritme. Insertion-Sort har

- $\Theta(n)$ i beste tilfelle
- $\Theta(n^2)$ i gjennomsnitt
- $\Theta(n^2)$ i verste tilfelle

Kan anvende Randomized-Select fra forelesning 4.

- $\Theta(n)$ i beste tilfelle
- $\Theta(n)$ i gjennomsnitt
- $\Theta(n^2)$ i verste tilfelle

K-LARGEST

Hvilke alternativer har vi til K-LARGEST?

Vi kan sortere A med en sorteringsalgoritme. INSERTION-SORT har

- $\Theta(n)$ i beste tilfelle
- $\Theta(n^2)$ i gjennomsnitt
- $\Theta(n^2)$ i verste tilfelle

Kan anvende RANDOMIZED-SELECT fra forelesning 4.

- $\Theta(n)$ i beste tilfelle
- $\Theta(n)$ i gjennomsnitt
- $\Theta(n^2)$ i verste tilfelle

Kan anvende SELECT fra forelesning 4.

- $\Theta(n)$ i beste tilfelle
- $\Theta(n)$ i gjennomsnitt
- $\Theta(n)$ i verste tilfelle

Dual-Sort

Hva er kjøretiden til Dual-Sort**?**

DUAL-SORT

Hva er kjøretiden til DUAL-SORT?

$\lfloor \frac{n}{2} \rfloor$ iterasjoner hvor det utføres $n - 2i$ operasjoner i iterasjon i.

Hva er kjøretiden til DUAL-SORT?

$\lfloor \frac{n}{2} \rfloor$ iterasjoner hvor det utføres $n - 2i$ operasjoner i iterasjon i.

$$\sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} (n - 2i)$$

Hva er kjøretiden til Dual-Sort?

$\lfloor \frac{n}{2} \rfloor$ iterasjoner hvor det utføres $n - 2i$ operasjoner i iterasjon i.

$$\sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} (n - 2i) = \left\lfloor \frac{n}{2} \right\rfloor n - \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} 2i$$

DUAL-SORT

Hva er kjøretiden til DUAL-SORT**?**

$\left\lfloor \frac{n}{2} \right\rfloor$ iterasjoner hvor det utføres $n - 2i$ operasjoner i iterasjon i.

$$\sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} (n - 2i) = \left\lfloor \frac{n}{2} \right\rfloor n - \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} 2i \approx \frac{n^2}{2} - \frac{n^2}{4} - \frac{n}{2}$$

Hva er kjøretiden til Dual-Sort?

$\lfloor \frac{n}{2} \rfloor$ iterasjoner hvor det utføres $n - 2i$ operasjoner i iterasjon i.

$$\sum_{i=1}^{\lfloor \frac{n}{2} \rfloor}(n-2i) = \left\lfloor \frac{n}{2} \right\rfloor n - \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} 2i \approx \frac{n^2}{2} - \frac{n^2}{4} - \frac{n}{2} = \frac{n^2}{4} - \frac{n}{2}$$

Hva er kjøretiden til Dual-Sort?

$\lfloor \frac{n}{2} \rfloor$ iterasjoner hvor det utføres $n - 2i$ operasjoner i iterasjon i.

$$\sum_{i=1}^{\lfloor \frac{n}{2} \rfloor}(n-2i) = \left\lfloor \frac{n}{2} \right\rfloor n - \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} 2i \approx \frac{n^2}{2} - \frac{n^2}{4} - \frac{n}{2} = \frac{n^2}{4} \; - \frac{n}{2} = \Theta(n^2)$$

Dual-Sort

Hva er kjøretiden til Dual-Sort**?**

$\lfloor \frac{n}{2} \rfloor$ iterasjoner hvor det utføres $n - 2i$ operasjoner i iterasjon i.

$$\sum_{i=1}^{\lfloor \frac{n}{2} \rfloor}(n - 2i) = \left\lfloor \frac{n}{2} \right\rfloor n - \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} 2i \approx \frac{n^2}{2} - \frac{n^2}{4} - \frac{n}{2} = \frac{n^2}{4} \ - \frac{n}{2} = \Theta(n^2)$$

Hvordan beviser vi at Dual-Sort **stemmer?**

DUAL-SORT

Hva er kjøretiden til DUAL-SORT?

$\lfloor \frac{n}{2} \rfloor$ iterasjoner hvor det utføres $n - 2i$ operasjoner i iterasjon i.

$$\sum_{i=1}^{\lfloor \frac{n}{2} \rfloor}(n - 2i) = \left\lfloor \frac{n}{2} \right\rfloor n - \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} 2i \approx \frac{n^2}{2} - \frac{n^2}{4} - \frac{n}{2} = \frac{n^2}{4} - \frac{n}{2} = \Theta(n^2)$$

Hvordan beviser vi at DUAL-SORT stemmer?

Bruker induksjon

Dual-Sort

Hva er kjøretiden til $\textsc{Dual-Sort}$**?**

$\lfloor \frac{n}{2} \rfloor$ iterasjoner hvor det utføres $n - 2i$ operasjoner i iterasjon i.

$$\sum_{i=1}^{\lfloor \frac{n}{2} \rfloor}(n - 2i) = \left\lfloor \frac{n}{2} \right\rfloor n - \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} 2i \approx \frac{n^2}{2} - \frac{n^2}{4} - \frac{n}{2} = \frac{n^2}{4} \ - \frac{n}{2} = \Theta(n^2)$$

Hvordan beviser vi at $\textsc{Dual-Sort}$ **stemmer?**

Bruker induksjon

Løkkeinvariant: Etter iterasjon i består $\mathrm{A}[1..i]$ av de i minste tallene i A og $\mathrm{A}[\mathrm{A}.\mathit{length} - i + 1..\mathrm{A}.\mathit{length}]$ av de i største tallene i A. Begge er sortert i stigende rekkefølge.