

Instructions

TDT4121 – Innføring i algoritmer

Øvingsforelesning, 31/08

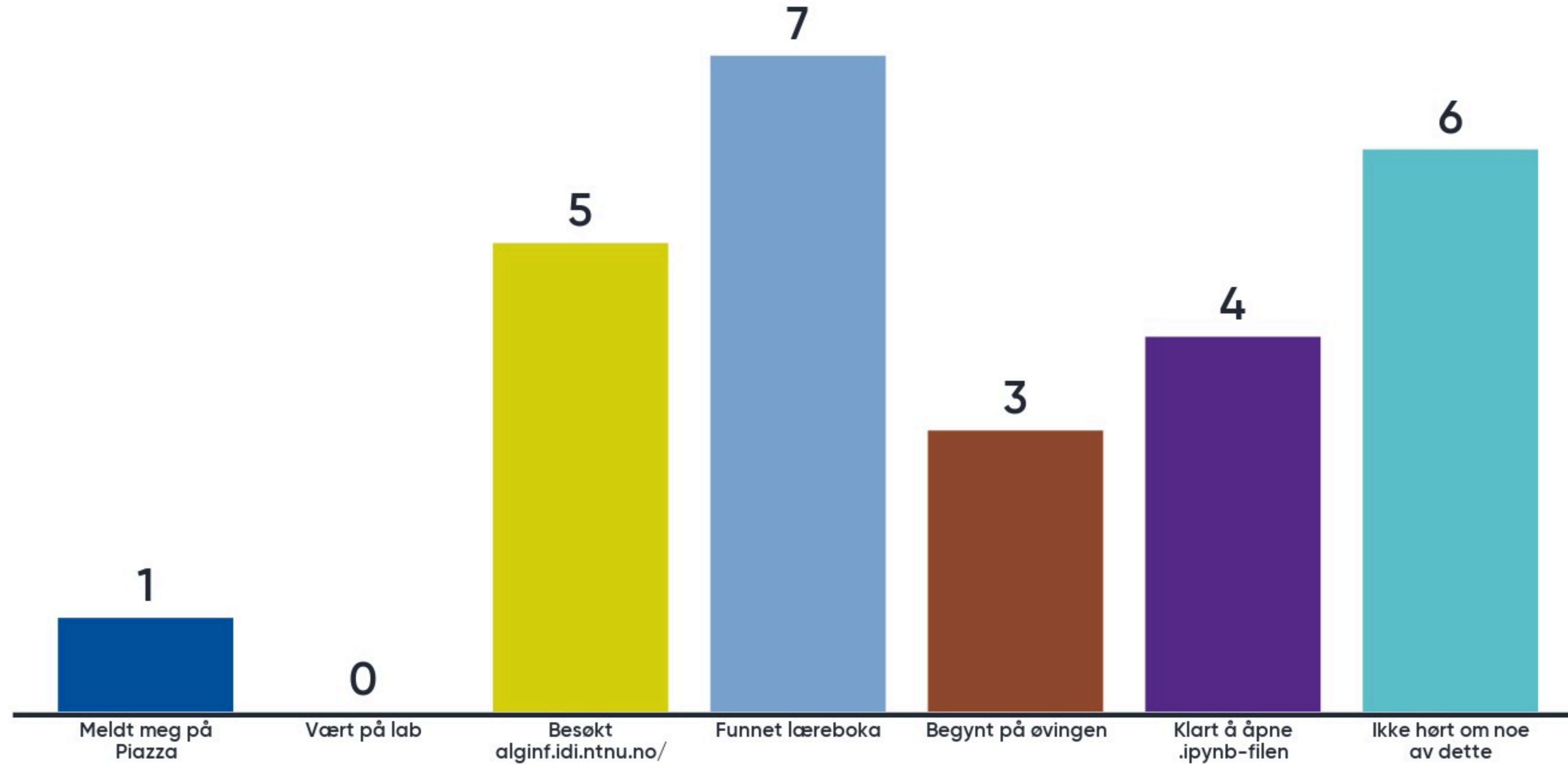
Agenda

- Hei!
- Informasjon/Spørsmål
- Kort introduksjon til kap. 1 og 2

Undervisningsassistenter

- Andre Omdahl, andreomd@stud.ntnu.no
- Jacob Fagerlund Hambro, jacobfh@stud.ntnu.no
- Magne Tenstad, magneet@stud.ntnu.no
- Nicoline Mork, nicolimo@stud.ntnu.no
- Sivert Utne, sivertut@stud.ntnu.no

Kryss av de som stemmer. Jeg har..



Piazza

- Spørsmålsforum
- Nås gjennom Blackboard
- Mulig å være anonym
- Vi svarer så fort vi har mulighet

Lab

- Få hjelp av studentassistentene
- Hverdager, kl 10 – 16*
- Drop-in, ingen påmelding*
- *kan bli endret



alginf.idi.ntnu.no

- Informasjon om forelesninger, øving og pensum
- Sjekk her før du spør på Piazza!

Jupyter Notebook (.ipynb)

- Python og Markdown i samme fil
- Utgjør programmeringsdelen av øvingene
- Åpnes i JupyterHub (jupyterhub.apps.stack.it.ntnu.no)
- ..eller i VSCode med Jupyter Extension

Øvingsforelesninger

- Onsdager, kl 15:15–17:00
- Vi ønsker å være til nytte for dere
- Hovedoppgave: Forklare løsningsforslag til øvingene
- Andre muligheter: Repetere pensum, Spørsmål og svar, Quiz, ...

Hva slags øvingsforelesninger ønsker dere?

interaktive gjennomganger av øvingene

Kok 😎

Kok 😎

Eksempeloppgaver relevant for kommende øving

Se på eksempler som er relevante til neste ukes øving

Mulig med opptak? Timeplan-kræsje

Litt nyttige tips og forklaringer til øvingene

Forklaring av "vanskelige" oppgaver, grundig gjennomgang.

Litt repetering av pensum, gjennomgå fasit og tips og triks til kommende innleveringer

Hva slags øvingsforelesninger ønsker dere?

Kode løsningsforslaget

Repetisjon av pensum

Gjennomgå de vanskeligste oppgavene fra øvinger som har vært, gå igjennom liknende oppgaver som i øvingene som ikke har hatt frist

Noe relevant til den aktuelle øvingen/
eksamen

Praktisk utførelse/koding er mest naturlig i øvingsforelesningene :)

Spørsmål?

0 questions
0 upvotes

I pausen

- (om du ikke allerede har gjort..)
- Meld deg på Piazza
- Sjekk ut alginf.idi.ntnu.no
- Åpne Jupyter Notebook



Kap. 1 og 2

1. Introduction: Some Representative Problems*
2. Basics of Algorithm Analysis**

*Stabil matching

**Asymptotisk notasjon

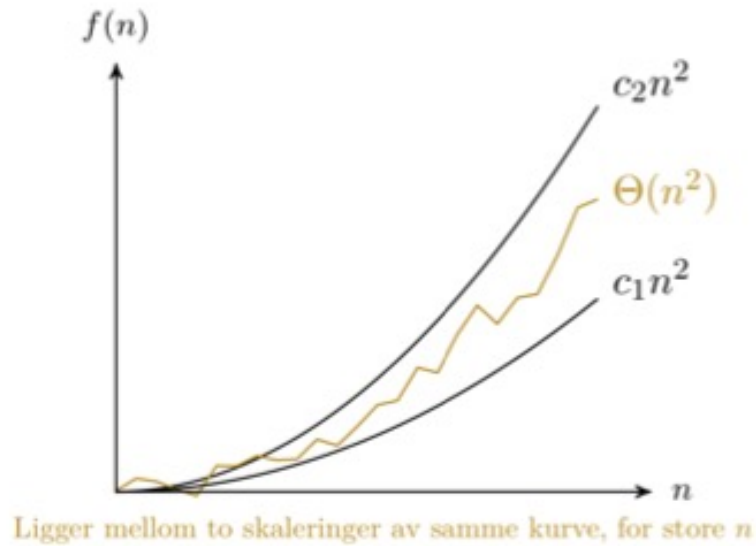


Hva er en algoritme?

- En algoritme er en definert beregningsprosedyre med endelig antall steg.
- Tar inn en eller flere verdier som input og produserer en eller flere verdier som output

Asymptotisk notasjon

Θ -notasjonen



O -notasjonen



Ω -notasjonen



- Kunne gi enkel og grov beskrivelse av utviklingen til funksjoner
- Beskriver en asymptotisk grense for en funksjon, ved hjelp av en enklere funksjon

💡 $O(g(n)) = \{f(n): \text{det eksisterer positive konstanter } c \text{ og } n_0 \text{ slik at } 0 \leq f(n) \leq cg(n) \text{ for alle } n \geq n_0\}$

💡 $\Theta(g(n)) = \{f(n) : \text{det eksisterer positive konstanter } c_1, c_2 \text{ og } c_3 \text{ slik at } 0 \leq c_1g(n) \leq f(n) \leq c_2g(n) \text{ for alle } n \geq n_0\}$

$\Omega(g(n)) = \{f(n): \text{det eksisterer positive konstanter } c \text{ og } n_0 \text{ slik at } 0 \leq cg(n) \leq f(n) \text{ for alle } n \geq n_0\}$

$O(3 * n^2 + 10n + 10)$ blir $O(n^2)$

Fjern konstanter og ledd av lavere orden

$$O(\cancel{n} + 10) \qquad O(n)$$

$$O(1000\cancel{n} + 50) \qquad O(n)$$

$$O(n^2 + \cancel{5n} + 8) \qquad O(n^2)$$

Hva mener vi med å analysere algoritmer?

- Mest vanlig er *beregningstiden (kjøretiden)*
 - Antall primitive operasjoner , eller "steg", som utføres
 - Kjøretiden til en algoritme er summen av kjøretidene til hver påstand som blir utført
- Andre ressurser er minne, bredbåndbredde for kommunikasjon eller maskinvare (hardware)

Algorithm <i>arrayMax</i> (<i>A</i> , <i>n</i>)	# operations
<i>currentMax</i> $\leftarrow A[0]$	2
for <i>i</i> $\leftarrow 1$ to <i>n</i> - 1 do	$2n$
if <i>A</i> [<i>i</i>] > <i>currentMax</i> then	$2(n - 1)$
<i>currentMax</i> $\leftarrow A[i]$	$2(n - 1)$
{ increment counter <i>i</i> }	$2(n - 1)$
return <i>currentMax</i>	1
	Total $8n - 2$