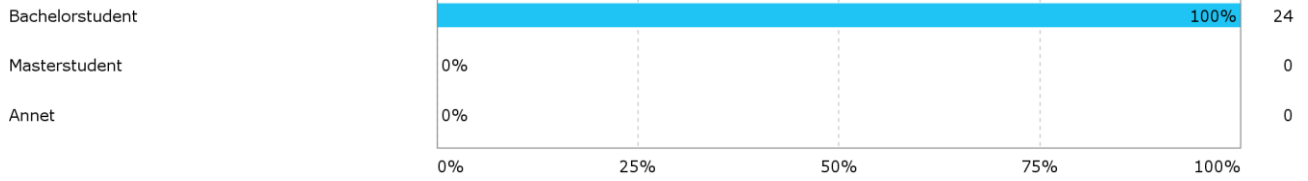


# INF102

## Er du?

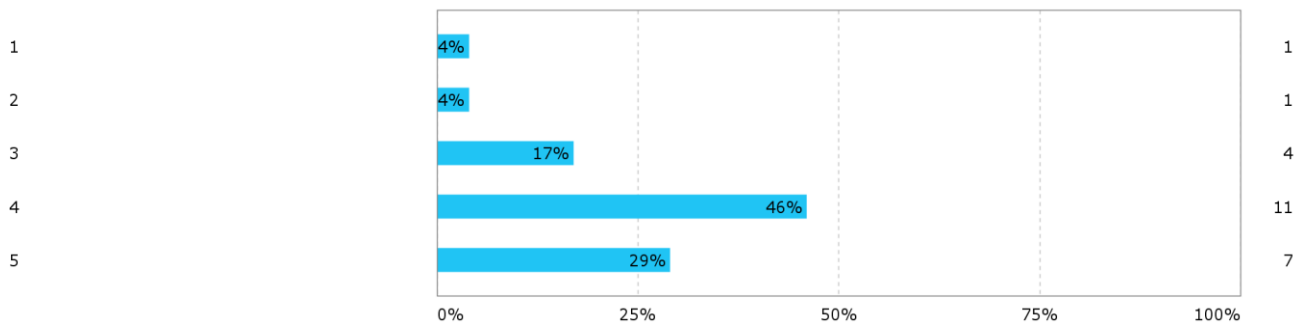


## Er du? - Annet

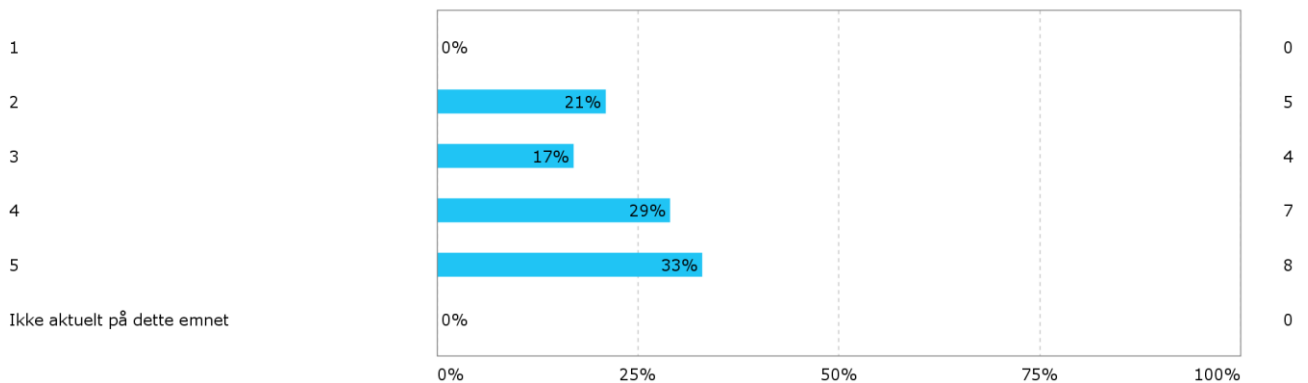
Hvor mye gjennomsnittlig arbeidstid per uke har du brukt på dette emnet (inkludert forelesninger, gruppeøvelser, lab/felt, egenstudier)?

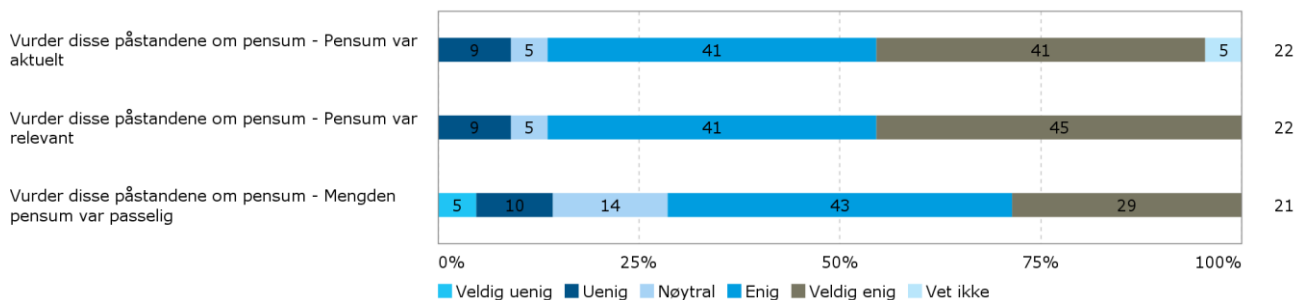
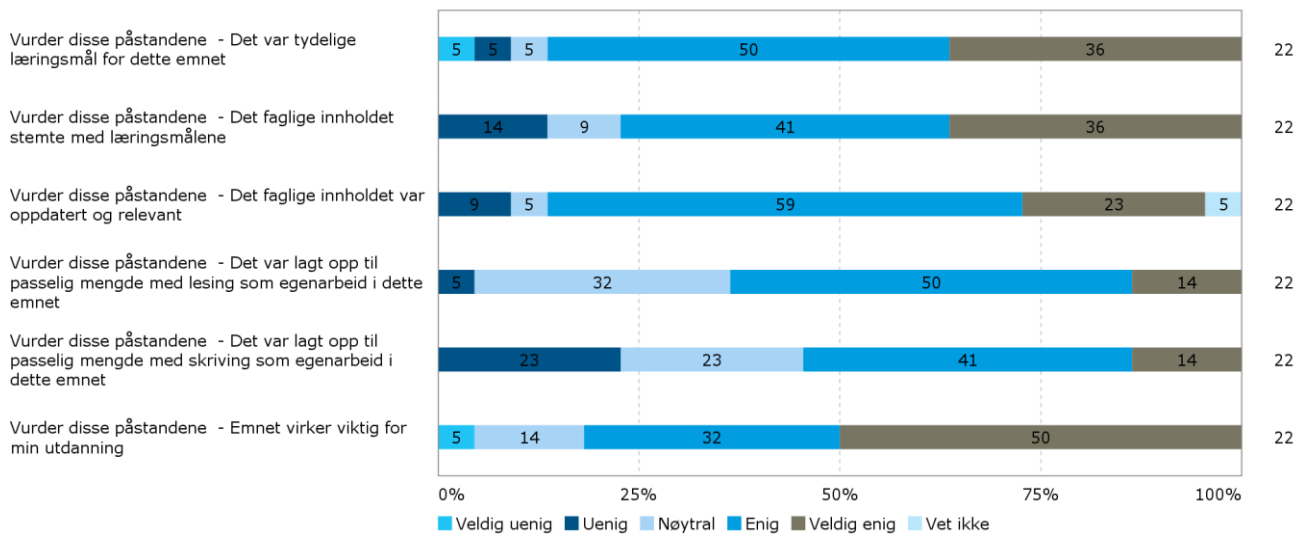


Hvor mye teoretisk kunnskap har du tilegnet deg på dette emnet? (1 = ingen, 5 = mye)

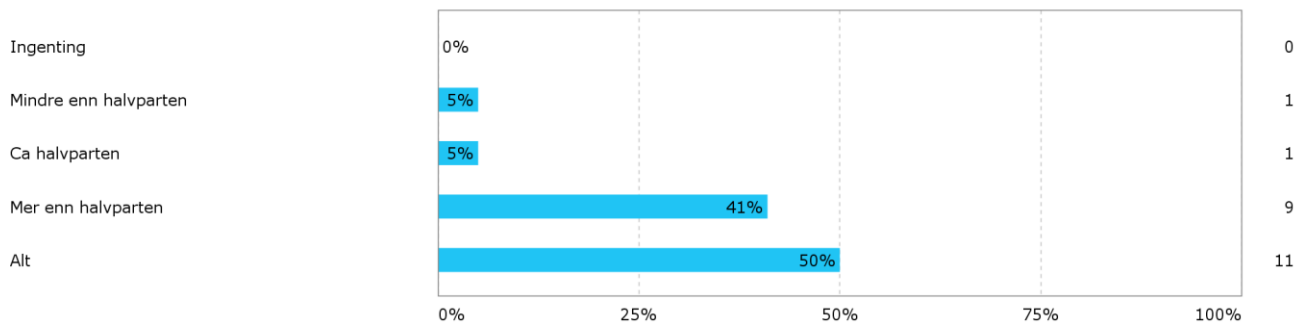


Hvor mye praktisk kunnskap har du tilegnet deg på dette emnet? (1 = ingen, 5 = mye)

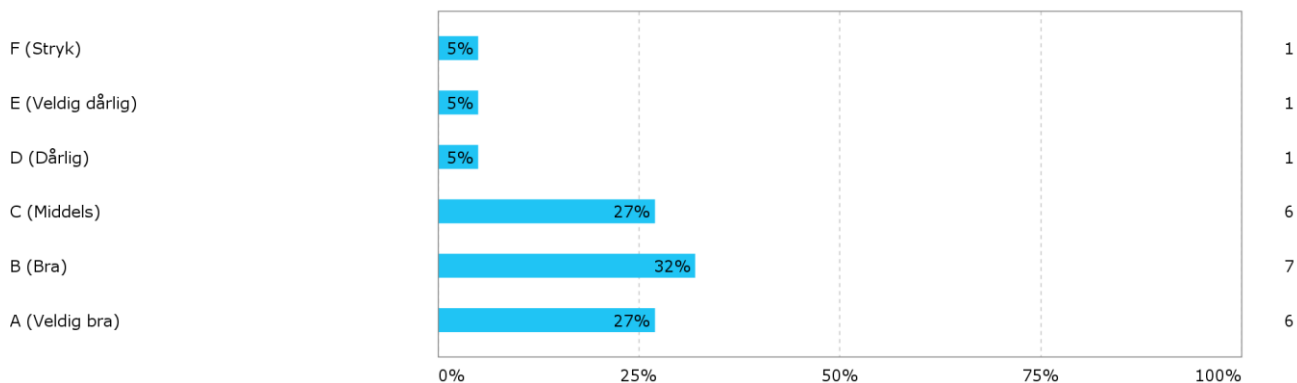




### Hvor mye av pensum leste du?



### Hvilken karakter vil du gi dette emnet?



## Hva likte du mest med dette emnet?

- Foreleser
- Emnet er lærerikt, og deler er svært interessant
- Å lære om trær og grafer. Å se hvor mye mer effektivt man kan lage et program
- Spennende tema
- Læreboken og tillegsmateriell var av veldig god kvalitet og veldig faglig relevant
- det var spennende å jobbe med obligatoriske oppgaver.  
Bra med ukentlige oppsummeringstimer på fredager, flink gruppeleder
- Flink og engasjert foreleser
- At det gav mer innsikt i programmering generelt, sitter igjen etterpå med et litt større bilde.
- Flink foreleser
- Spennende, får deg til å tenkte nytt om programmering
- God blanding av teori og praksis i form av implementasjon og bruk av de forskjellige algoritmene.  
Likte også at alle eksempler ble lagt ut på github så man enkelt kunne laste dem ned.
- Pensum, forelesingane, læreboka (+nettsida!)

## Hva likte du minst med dette emnet?

- Obligene var for stor
- Kravene på obligene var kanskje litt strenge til tider og at en ikke vet hva en kunne få på eksamen (men det skyldes jo at tidligere eksamener hadde et mer teoretisk fokus)
- Fikk aldri helt grep om kjøretidsanalyse. Klarte heller ikke å nytte meg godt av gruppen
- For lite teori
- obligatoriske øvingene hadde varierende frister og litt uklare rammer for hvordan grensesnittet til programmet man laget skulle være
- Altfor mange innleveringer: siste frist var to uker før eksamen - og da er det siste sjanse å melde seg fra eksamen hvis man ikke får bestått siste innlevering (dårlig planlegging med obligatoriske oppgaver).  
Det var overaskende at det var bare en type spørsmål på eksamen - skrive (pseudo)kode, man kunne ikke tenke at det å kunne kode er sentralt i emnet (og ikke selve algoritmer)
- eksamensformen
- Eksamen kanskje, var vanskelig å vite hvordan oppgavene kom til å være, om det ble spørsmål og diskusjon eller direkte programmeringsoppgaver. Krever ulik forberedelse.
- Arbeidsmengden i de obligatoriske oppgavene.
- Eksamen
- Ikke syntax highlightning på eksamen, selv om inspera støtter dette.
- Enkelte av dei obligatoriske oppgåvene var noko uklare - óg for gruppeleiarane
- Foreleseren og eksamen var lagt opp

## Har du forslag til hvordan emnet kan forbedres?

- Minske oblig størrelsen
- Finne en måte å illustrere kjøretidsanalyse på
- Inkluder abstrakt pseudokode av algoritmane som det er enkelt å resonnerer eigenskapene til algoritmane ut frå.

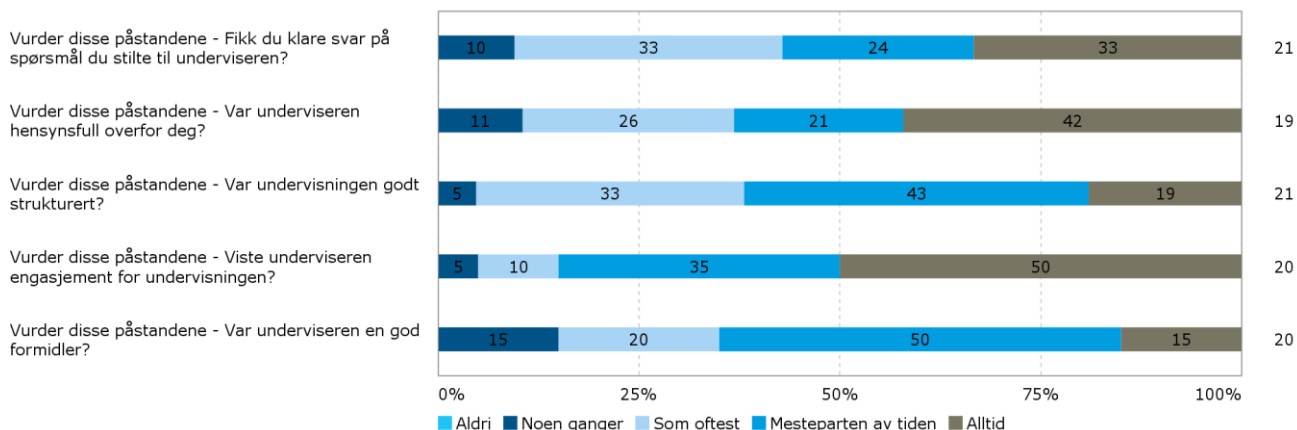
Vurdering bør ta for seg dei ulike læringsmålene i kvar sine oppgåver slik at det er mogleg å få til nokre læringsmål sjølv om andre læringsmål kanskje er mangelfulle. Kjøretidsanalyse kan med fordel vere knytt til kode som er oppgitt i staden for til eigen kode, og noko teori bør også vere med i både pensum og eksamen.

- fikse opp litt i det obligatoriske øvingsopplegget, f.eks. ha flere test input og output der man skal kode algoritmer, kanskje ha et basic grensesnitt opp for programmet definert, evt. laste opp kode som kjører mot testcases for å få det godkjent
- Planlegge bedre innleveringer - i god tid før eksamen (dere kan vurdere å gi to obligatoriske oppgaver, ikke tre, så blir det mindre stress og panikk rett før eksamen).  
Kan også gjøre disse vinduene der vi må skrive i spørreundersøkelse litt større - jeg kan se bare ett ord om gangen når jeg skriver ( har dårlig oversikt over det som jeg skrev akkurat nå)
- La Marc få muligheten til å forelese faget neste år igjen, da tror jeg det bare kan bli bedre.
- Mer bruk av animasjoner som viser stegene algoritmen bruker, istedenfor alltid å tegne, dette passer godt til de mest heftige algoritmene, som f.eks `tree.swim()` , `tree.delete()` , `graph.findCycle()` og kanskje ta det for binary sort, siden det er første search algoritmen og derfor er viktig at folk faktisk skjønner den veldig godt!

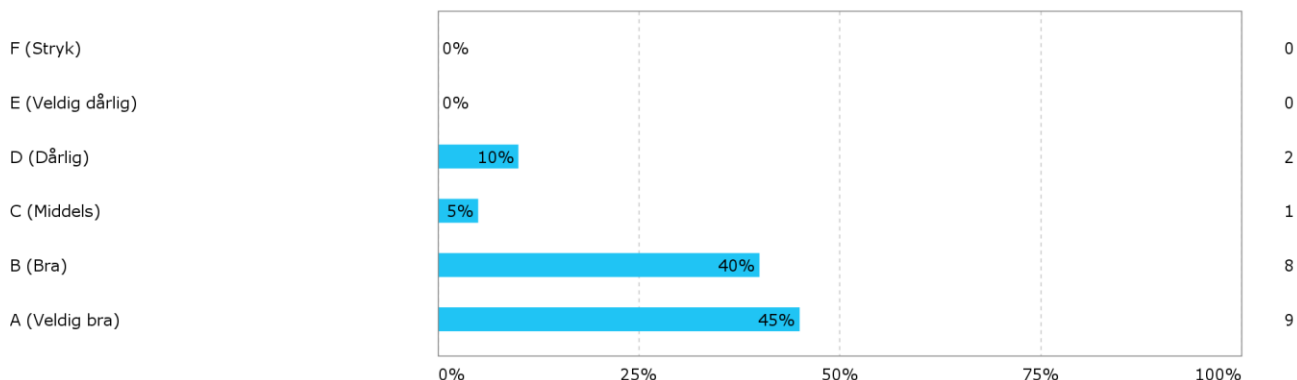
- Likte faget godt! Eit forbedringspotensiale: Gjere det veldig tydeleg i oppgåvene kva ein forventer at studenten skal gjere, og sørge for at alle gruppeleiarane forstår oppgåvene likt.
- Klare mål

## Tilbakemeldinger på organisert praktisk undervisning?

- Gruppeleder hadde flere ganger ikke lest oppgaven og viste ikke helt hva den gikk ut på. Blir da vanskelig og hjelpe oss.
- Gruppelederne har vært veldig hjelpsomme, ingen ting å utsette der.
- Gruppeundervisningen var alt for lite strukturert. Trengte en tydeligere pekepinn på hvordan jeg burde benyttet meg av gruppen
- Det er uheldig at det er lov å bruke standard javabibliotek, og læreboka sitt API på veke-oppgåvene når dette ikkje er lov på eksamen. Dvs at me bør få klar beskjed tidleg, om at ein må klare dette utan bruk av desse APIene på eksamen dersom det er ein viktig del av kurset. Alternativt at pensum endres slik at ein har lov til å bruke denne koden på eksamen, bortsett frå t.d. ein oppgåve der ein t.d. skal implementere ein stabel, eller ein heap.
- Flott med fredags oppgavegjennomgang.  
Var ikke så mye i grupper (lite hjelp fra gruppeledere - de forutsetter at man kan mye mer enn jeg faktisk kunne)
- Veldig bra, gode forklaringer av foreleser med illustrasjoner på tavle osv. Syns kanksje gruppelederne kunne vært litt mer forberedt på oppgavene slik at de visste hva oppgavene gikk ut på når man spurte om hjelp. Trengte hjelp 2-3 ganger og alle gangene visste ikke gruppeleder (forskjellige) hva oppgaven var, så den måtte forklares/diskuteres først. Unødvendig bruk av tid. Ellers bra hjelp når hjelpen kom!
- Godt med obligatoriske oppgaver, merket at mange medelever ikke gjorde ukesoppgaver fordi de mente det ble for mye tulleoppgaver som ikke var relevant for eksamen. Hvis du tar og korter ned på oppgavene og formidler at disse oppgavene virkelig ER viktige for eksamen, vil det bli mye mindre skippertak på mange.
- Gode og aktuelle for emnet.
- Gruppeøvelser: Gruppeleiarane var flinke og var til god hjelp!



## Hvilken karakter vil du gi underviseren(e)?

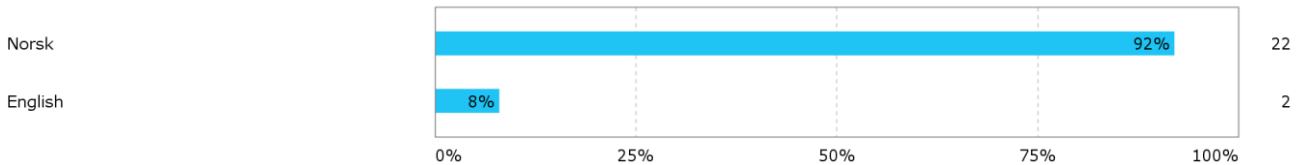


## Har du forslag til hvordan underviseren kan forbedre sin undervisning?

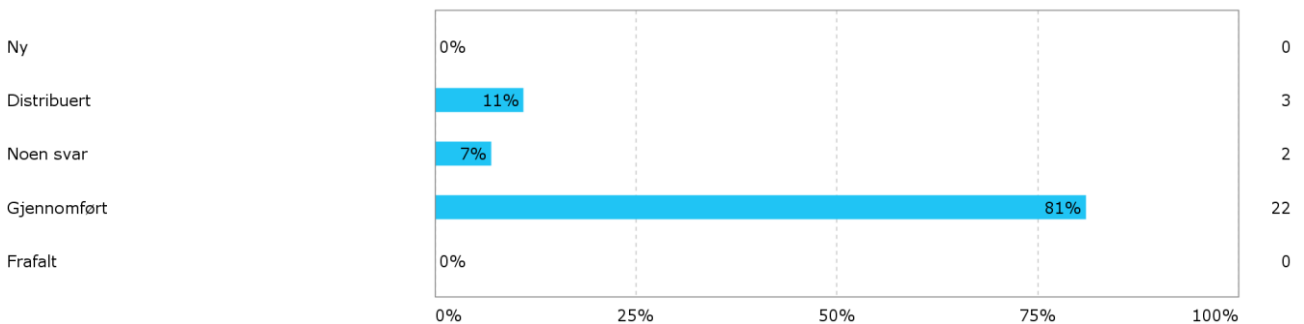
- Bättre abstrakt pseudokode-gjennomgang av algoritmene, til dels uavhengig av datastrukturer, før ein gjennomgår algoritmene med bruk av datastrukturene.
- Av og til litt overfokus på detaljer eller forvirrende traces av algoritmer - mye av dette var bra, men noen ganger følte jeg at tråden ble litt mistet og tracene tok for lang tid relativ til diskusjon rundt konseptene bak algoritmene
- Mer animasjoner for å forstå, mer konkrete ukesoppgaver, konkurransen på oblig 2 var kjempebra!

Bra med mye diskusjon i timen.

## Språk



## Samlet status



## Kommentar fra underviser:

Evalueringen er for det meste positiv, og siden jeg også bli tildelt foreleserprisen av studentene er det ingen grunn til bekymring. Forslagene til hvordan emnet kan forbedres er konstruktive. Her kommer de viktigste (som ikke alle er lett å gjennomføre):

1. Vi må prøve å få tak i de allerbeste gruppeledere for grunnevrer som 100, 101 og 101. Koordinator er også viktig, men det virker å ha fungert veldig bra.
2. Vi kan eventuelt legge litt mindre vekt på programmering, og litt mer på teori. Egnetlig overraskende at noen studenter nevner det, det er litt uklar hva de mener med teori. Kanskje kjøretidsanalyse (som er nokså vanskelig), eller fakta om algoritmer?
3. Et interessant forslag er å finne andre måter å presentere algoritmenes virkemåte på, som f.eks. animasjon. Dette er egentlig et fint tema for et 319 prosjekt, eller t.o.m. en kort masteroppgave. Dette skal jeg prøve å følge opp.
4. Planlegging av siste oblig er gjort vanskelig av fargekodesystemet som fakultetet bruker: eksamen kommer veldig tidlig etter siste (grønne) periode for innlevering av oblig. Dette forverres av en enda tidligere frist for godkjentlisten. Med eksamen 27.11 blir alt dette veldig vanskelig.

5. Eksamensformen kan justeres etter nye muligheter som Inspira kommer til å tilby, forhåpentligvis bruk av kompilator.