Se instruksjoner på side 4.

[05] 1. Dersom $f(n) = \Theta(g(n))$ og $g(n) = \Theta(h(n))$ vet vi at $f(n) = \Theta(h(n))$

[05] 2. Løs rekurrensen $T(n) = 3T(n/3) + 3$: $T(n) = \Theta($ n $)$

[05] 3. Løs rekurrensen $T(n) = T(\sqrt{n}) + \lg n$: $T(n) = \Theta($ $\lg n$ $)$

[05] 4. Radix sort kan sortere n tall med d siffer i området $1 \ldots k$ med kjøretid $\Theta(d(n + k))$

[05] 5. Søk i en lenket liste med n elementer har kjøretid $\mathcal{O}(n)$

[05] 6. En tilsynelatende tilfeldig hashfunksjon er ment å minimere antall kollisjoner

[05] 7. Hvis 101 personer tar hverandre i hånden blir det 5050 håndtrykk totalt

[05] 8. Dijkstras algoritme er korrekt dersom grafen har positive vekter

[05] 9. Dynamisk programmering gir ytelsesbesparelser når vi har overlappende delproblemer

[05] 10. Hvis vi øker kapasiteten til hver enkelt kant i et flytnettverk med k, vil verdien til den maksimale flyten alltid øke med et multiplum av k? Begrunn svaret kort.

> Nei. Det holder å sette opp et moteksempel.

[07] 11. Anta at du har en tom binær max-heap. Sett inn følgende verdier i rekkefølge, én etter én:

[2, 4, 5, 9, 1, 10, 7, 3, 8, 6]

Utfør deretter HEAP-EXTRACT-MAX to ganger på heapen. Anta at du representerer heapen som en tabell. Hvordan ser den ut nå?

> [8, 6, 7, 5, 3, 4, 1, 2]

[07] 12. Hva gjør denne algoritmen?

```
for α = 1...ω
    for β = 1...ω
        for γ = 1...ω
            π_βγ = max(π_βγ, π_βα · π_αγ)
```

> Modifikasjon av FLOYD-WARSHALL (men det kreves ikke at man nevner det, spesifikt). Finner stier med størst produkt av kantvekter mellom alle nodepar.
>
> (Vil bare være korrekt så lenge ingen sykel har et produkt større enn 1 – ekvivalent med en negativ sykel i korteste-vei–problemet. Det kreves ikke at kandidaten påpeker dette.)
>
> Om man antar at variablene er boolske, så vil algoritmen finne transitiv tillukning (*transitive closure*); dette er også et akseptabelt svar.

[08] 13. Anta at du har n geografiske regioner, der mange regioner overlapper andre, og der hver region er definert ved et sett med $\mathcal{O}(1)$ punkter. Anta at du har en prosedyre som kan kombinere to regioner med hhv. k og ℓ punkter til en ny region som er unionen av disse, og som defineres av maksimalt $k + \ell$ punkter, og at denne prosedyren har en kjøretid på $\mathcal{O}(k + \ell)$.

Du ønsker å kombinere alle regionene. Hvordan vil du gjøre dette? Hva blir kjøretiden?

Splitt og hersk: Del i to mengder, kombinér regionene i hver mengde rekursivt, og kombinér i lineær tid. Total kjøretid $\Theta(n \log n)$.

Evt. kan man hele tiden kombinere de to minste (i praksis Huffmans algoritme). Samme kjøretid.

[08] 14. I en graf ønsker du å starte i en node u og ende opp i en annen node v, og besøke hver av de andre nodene nøyaktig én gang på veien. Vis kort at dette er NP-hardt.

Kan oppdage en Hamilton-sykel ved å prøve alle kanter (v, u).

[10] 15. To ord er anagrammer av hverandre dersom de består av de samme tegnene, men i forskjellig rekkefølge. Hvert tegn må forekomme like mange ganger, og vi regner et ord som et anagram av seg selv. Anta at du har oppgitt en tekst, og et start-ord A og et slutt-ord B. Du ønsker å konstruere en sekvens med ord som starter med A og som slutter med B, med følgende krav: Hvis to ord X og Y står ved siden av hverandre i denne sekvensen, skal et anagram for X være brukt i samme setning som et anagram for Y. Beskriv svært kort en algoritme som avgjør om en slik sekvens eksisterer.

Lag en hashtabell der nøklene er ordene, med bokstavene sortert og verdiene er noder i en graf. For hvert par med ord i en setning legg inn en kant. Se deretter om det finnes en sti fra noden for A til noden for B.

[10] 16. En gjeng med riddere vil avgjøre hvem som er best til å fekte. Hver ridder skal gjennomføre k fektematcher med hver av de andre, og den som vinner flest matcher totalt kåres til vinner. På et tidspunkt etter at en del av matchene alt har blitt avholdt, lurer Lady Lurwicke på om hun har noen sjanse til å vinne, eller eventuelt komme på en delt førsteplass. Beskriv en algoritme som lar henne finne ut dette.

Hint: Hvor mange kamper kan Lady Lurwicke i beste fall vinne? Finnes det et mulig scenario der ingen andre vinner flere kamper totalt?

Max-flyt: Én node per par med riddere (utenom Lady Lurwicke) med én kant til hver av de to (med uendelig kapasitet). Kapasitet fra s til hvert par er antall gjenværende kamper mellom de to.

Hvis ridder u har vunnet z kamper, og Lady Lurwicke har vunnet x og har igjen y totalt, kan u vinne $x + y - z$ kamper uten å slå Lurwicke. Dette blir kapasiteten på kanten (u, t).

Om alle kanter fra s fylles opp betyr det at alle kampene kan utkjempes uten at noen vinner flere enn Lurwicke, og hun har dermed en sjanse til å vinne. Eller har hun ikke det.

Merk: Det er ikke sagt noe om at en fektekamp kan ende uavgjort, og det var heller ikke intensjonen, siden man da bare trivielt kan anta at ingen andre vil vinne noen av de gjenværende kampene. Siden det ikke ble eksplisitt sagt at dette ikke var en akseptabel løsning vil et slikt svar gi noe uttelling, men ikke full score.

Noen viktige punkter:

(*i*) Vennligst les hele eksamenssettet nøye før du begynner. Disponér tiden og forbered evt. spørsmål før faglærer kommer til eksamenslokalet.

(*ii*) Bruk gjerne blyant! Evt. kladd på eget ark først for å unngå overstrykninger.

(*iii*) Gjør antagelser der det er nødvendig. Skriv svarene dine på oppgavearket, som angitt, enten på linjer der det mangler tekst i oppgaven, eller i svarruter under oppgaveteksten. Hold svarene korte og konsise, om mulig. Lange forklaringer som ikke direkte besvarer spørsmålene tillegges liten eller ingen vekt.

(*iv*) Med mindre annet er oppgitt kan du beskrive algoritmene dine i prosa, pseudokode eller programkode, etter eget ønske, så lenge det kommer klart frem hvordan algoritmen virker. Korte, abstrakte forklaringer kan være like gode som utførlig pseudokode, så lenge de er klare og presise. I de fleste tilfeller vil en slik kort besvarelse kreve mindre arbeid og ha færre feilkilder.

(*v*) Algoritmene dine bør være så effektive som mulig, med mindre annet er oppgitt. Kjøretider gis i asymptotisk notasjon, så presist som mulig.

Eksamen har 16 oppgaver, totalt verdt 100 poeng. Poengverdi er angitt til venstre for hver oppgave.