**Noen viktige punkter:**

(*i*) Les hele eksamenssettet nøye *før* du begynner!

(*ii*) Faglærer går normalt én runde gjennom lokalet. Ha evt. spørsmål klare!

(*iii*) Skriv svarene dine i svarrutene og levér inn oppgavearket. Bruk gjerne blyant! Evt. kladd på eget ark først for å unngå overstrykninger, og for å få en egen kopi.

(*iv*) Ekstra ark kan legges ved om nødvendig, men det er meningen at svarene skal få plass i rutene på oppgavearkene. Lange svar teller ikke positivt.

Eksamen har 20 oppgaver, totalt verdt 120 poeng.
Av disse er 20 bonuspoeng, så en poengsum over 100 regnes som 100.
Poengverdi er angitt ved hver oppgave.

* * *

For hver av de første 5 oppgavene skal du velge svaret blant disse algoritmene, og sette kryss i sirkelen til venstre for rett bokstav. For hver oppgave kan det være snakk om én eller flere algoritmer, så det er tillatt å sette flere kryss.

A Bucket-Sort

B Counting-Sort

C Heapsort

D Insertion-Sort

E Merge-Sort

F Quicksort

G Radix-Sort

H Randomized-Quicksort

I Randomized-Select

J Select

(5 p) 1. Hvilke av algoritmene på 1 er baserte på designmetoden *divide-and-conquer*?

○ A ○ B ○ C ○ D ○ E ○ F ○ G ○ H ○ I ○ J

(5 p) 2. Hvilke av algoritmene på 1 har kjøretid $O(n)$ i *verste tilfelle*, dersom $k = O(n)$ og $d = O(1)$?

○ A ○ B ○ C ○ D ○ E ○ F ○ G ○ H ○ I ○ J

(5 p) 3. Hvilke av algoritmene på 1 bruker subrutinen PARTITION, evt. litt modifisert?

○ A ○ B ○ C ○ D ○ E ○ F ○ G ○ H ○ I ○ J

(5 p) 4. Hvilke av algoritmene på 1 benytter seg av en prioritetskø?

○ A ○ B ○ C ○ D ○ E ○ F ○ G ○ H ○ I ○ J

(5 p) 5. Hvilke av algoritmene på 1 endrer kjøretid fra beste til verste tilfelle?

○ A ○ B ○ C ○ D ○ E ○ F ○ G ○ H ○ I ○ J

(5 p) 6. Hva er løsningen på rekurrensen $T(n) = 3T(n/3) + n$? Oppgi svaret i O-notasjon.

(5 p) 7. Hva er løsningen på rekurrensen $T(n) = T(n/5) + n$? Oppgi svaret i O-notasjon.

(5 p) 8. Hva er løsningen på rekurrensen $T(n) = 2T(4\sqrt{n}) + \lg n$? Oppgi svaret i O-notasjon.

(5 p) 9. Hvis du setter verdiene 1, 2, 9, 5, 10, 7, 6, 4, 8 og 3 inn i et tomt binærtre (én etter én, i oppgitt rekkefølge), hva blir høyden til treet (antall kanter i den lengste stien fra rota til en løvnode)?

(5 p) 10. Hvis du setter verdiene 1, 2, 9, 5, 10, 7, 6, 4, 8 og 3 inn i en tom binær min-heap (én etter én, oppgitte rekkefølge), hva blir høyden til heapen (antall kanter i den lengste stien fra rota til en løvnode)?

(5 p) 11. I en sortert tabell med $n$ elementer forekommer et element $x$ én gang, mens alle andre elementer forekommer to ganger. Beskriv en effektiv algoritme for å finne ut hva $x$ er. Hva blir kjøretiden?

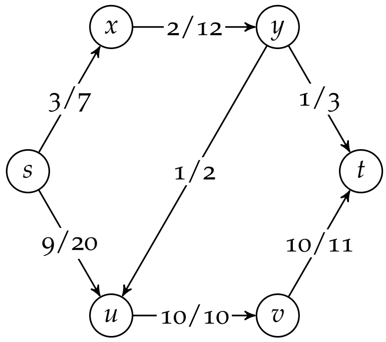
(5 p) 12. Forklar svært kort hva det vil si at en sorteringsalgoritme er *stabil*.

(5 p) 13. Du kjører DFS på en rettet graf, og får klassifisert alle kantene. Hvordan kan du bruke denne klassifiseringen til å avgjøre om grafen har en sykel?

(5 p) 14. Du har en binær matrise som angir om det går en veistrekning direkte mellom ulike par av byer. Du vil lage en ny matrise som angir om det går en direkte *eller indirekte* vei mellom hvert par, det vil si, hvis du får lov til å kjøre innom andre byer. Anta at det er snakk om $n$ byer. Hvordan vil du løse problemet hvis det er $\Theta(n)$ direkte veistrekninger? Hva blir kjøretiden?

(5 p) 15. Hvordan vil du løse problemet i oppgave 14 om det er $\Theta(n^2)$ direkte veistrekninger? Hva blir kjøretiden?

(5 p) 16. Tegn residualnettverket til flytnettverket som er tegnet i figur 1 (se s. 4).




Figur 1: Flytnettverk til oppgave 16.

(10 p) 17. Din venn Lurvik har funnet opp en ny algoritme som skal lage spenntrær for sammenhengende, uvektede, urettede grafer. Hun starter med en vilkårlig kant. Hun går så gjennom alle kantene igjen og igjen. I hver iterasjon legger hun til kanter som har nøyaktig én nabokant i spenntreet. (Kantene legges til én og én. For hver kant som legges til må man også ta hensyn til kantene som har blitt lagt til tidligere i samme iterasjon.) Er algoritmen korrekt? Forklar svært kort. Hvis grafen har $n$ noder og $m$ kanter, hva blir kjøretiden i verste tilfelle?

(**Merk:** Grafen er *uvektet*, så det er *ikke* snakk om å finne et *minimalt* spenntre, men et *vilkårlig* et.)

(10 p) 18. Anta at mengdene P og NP alt er definerte (dvs., du trenger ikke gi definisjoner for dem). Skissér svært kort definisjonen av NPC.

(10 p) 19. Du har en rad $v_1, \ldots, v_n$ med mynter, der $n$ er et partall. Du og en motspiller skal, annenhver gang, forsyne dere med enten den første eller siste mynten av dem som er igjen. Beskriv en algoritme som finner ut hvor mye du kan være *garantert* å vinne, dersom du begynner.

(10 p) 20. Du har oppgitt en graf der hver node er *rød* eller *grønn*, og det er ingen kanter mellom noder av samme farge. En grønn node har 0, 1 eller 2 naboer, mens de røde kan ha vilkårlig mange. Hver grønne node har også en positiv vekt. Du skal slette grønne noder, men den resulterende grafen må fortsatt ha stier mellom alle røde noder. Du ønsker å slette et sett med så høy totalvekt som mulig. Beskriv en algoritme som løser problemet, og forklar kort hvorfor svaret ditt blir riktig.