

## Løsning Repetisjonsøving i TDT4120

### Q1

Velg det alternativet som ordner de følgende kjøretidsklassene i stigende rekkefølge:

- linearitmisk  $O(n \log n)$
- eksponentiell  $O(2^n)$
- lineær  $O(n)$
- kubisk  $O(n^3)$
- kvadratisk  $O(n^2)$
- konstant  $O(1)$
- faktoriell  $O(n!)$
- logaritmisk  $O(\log n)$

Svar: konstant, logaritmisk, lineær, linearitmisk, kvadratisk, kubisk, eksponentiell, faktoriell

### Q2

Dersom  $f(n) = o(g(n))$  vet vi at

Svar:

- $g(n) = \Theta(f(n))$
- $g(n) = \omega(f(n))$
- $f(n) = O(g(n))$

### Q3

La tabellen  $S[1 \dots 10]$  være [ 50, 70, 45, 15, 72, 41, 61, 22, 26, 64 ] og la  $S.top = 5$ . Utfør så følgende operasjoner, i rekkefølge:

```
x = Pop(S)
y = Pop(S)
Push(S, x)
Push(S, y)
```

Hva er innholdet i tabellen  $S$  nå?

**\*\*Merk:\*\*** Det spørres her om innholdet i hele tabellen, ikke bare i stakken.

Svar:  $S = [ 50, 70, 45, 72, 15, 41, 61, 22, 26, 64 ]$

### Q4

Du får oppgitt at  $x.key = m$  og  $h(m) = j$  der  $h$  er en hashfunksjon. Da er...

Svar:  $x$  elementet,  $m$  nøkkelen og  $j$  hashen.

**Q5**

La  $T(n) = 27 * T(n/3) + n^2$ . Hvilket tilfelle tilhører rekurrensen når du benytter masterSvar: teoremet?

Svar: Case 1

**Q6**

Hva blir kjøretiden til funksjonen  $f(n)$ ? Anta at `println` tar konstant tid.

```
function f(n)
  for i = 1:n
    for j = i:n
      println("Dette er god eksamenstrening!")
    end
  end
end
```

Svar:  $\Theta(n^2)$

**Q7**

Anta at du har en sammenlikningsbasert sorteringsalgoritme  $S$  som sorterer heltallene i en tabell med lengde  $n$ . Hvilke(t) alternativ er korrekt?

Svar:

- $S$  kan ha best kjøretid  $\Theta(n)$ .
- $S$  kan være Quicksort.

**Q8**

Hvilke(n) av de følgende algoritmene er basert på designmetoden `divideSvar: andSvar: conquer`?

Svar:

- MergeSvar: Sort
- Quicksort
- RadixSvar: Sort
- RandomizedSvar: Quicksort
- RandomizedSvar: Select
- Select

**Q9**

Om man bruker BucketSvar: Sort på  $n$  uavhengig uniformt fordelte tall i området  $[0, 1)$ , så får man en kjøretid på  $O(n)$ . Underveis, som en del av BucketSvar: Sort, kalles InsertionSvar: Sort flere ganger. Hva er den forventede kjøretiden til hvert enkelt av disse kallene til InsertionSvar: Sort? (Det er her altså snakk om en forventningsverdi.) Bruk OSvar: notasjon og uttrykk svaret som funksjon av  $n$ .

**\*\*Merk:\*\*** Du skal ikke oppgi svaret som funksjon av inputSvar: størrelsen til InsertionSvar: Sort, men som funksjon av inputSvar: størrelsen til BucketSvar: Sort.

Svar:  $O(1)$

**Q10**

Du har oppgitt en tabell  $A = [1, 7, 2, 3, 5, 4, 6]$ .

Utfør BuildSvar: MaxSvar: Heap( $A$ ) og deretter HeapSvar: ExtractMax( $A$ ).

**\*\*Merk:\*\*** I svaret så er  $A$ .heapSvar: size =  $A$ .length – 1. Det spørres her om hele  $A$ , ikke bare heapen.

Svar:  $A = [6, 5, 4, 3, 1, 2, 2]$

**Q11**

Hvilke to egenskaper ved et problem ser vi etter for å avgjøre om vi vil bruke dynamisk programmering (DP)? Oppgi først egenskapen som er nødvendig for at DP skal være mer effektivt enn naturlige alternativer, og deretter egenskapen som er nødvendig for at DP skal gi korrekt svar.

Svar: Overlappende delproblemer og optimal substruktur/delstruktur

**Q12**

Merge-sort er et eksempel på en algoritme som rekursivt løser et problem. Hvorfor kan vi ikke bruke dynamisk programmering til å forbedre den?

Svar: Fordi delproblemene ikke overlapper

**Q13**

Hvilke to egenskaper må et problem ha for at vi kan bruke en grådig algoritme?

Svar: Grådighetsegenskapen og optimal substruktur

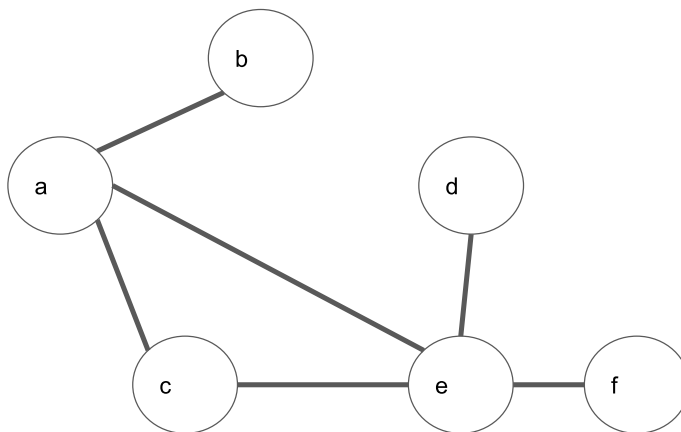
**Q14**

Hva har grådige algoritmer og dynamisk programmering til felles?

Svar: Begge utnytter optimal delstruktur

**Q15**

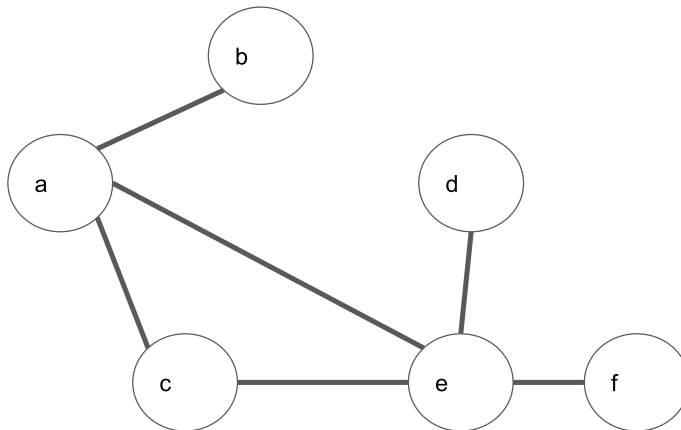
Hvilken rekkefølge blir nodene farget svart av et BFS søk på følgende graf med c som startnode? Anta at nabolisten til hver node er leksikografisk sortert.



Svar: c, a, e, b, d, f

**Q16**

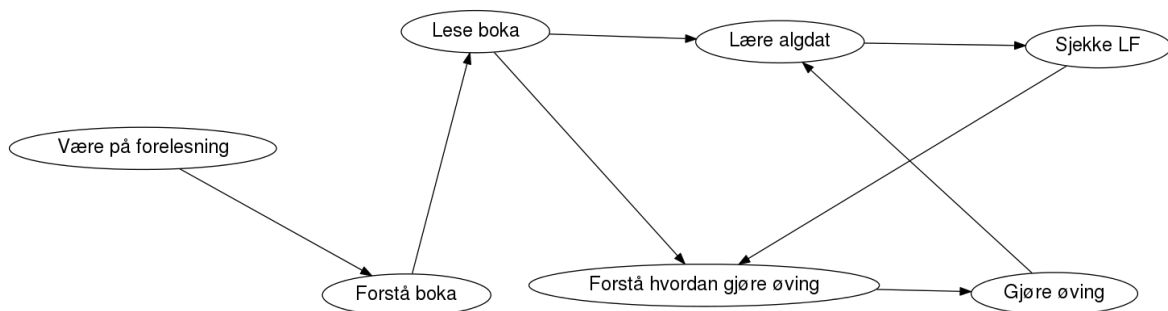
Hvilken rekkefølge blir nodene farget svart av et DFS søk på følgende graf med c som startnode? Anta at nabolisten til hver node er leksikografisk sortert.



Svar: b, d, f, e, a, c

**Q17**

Finn en topologisk sortering av følgende graf



Svar: Det finnes ingen topologisk sortering fordi grafen ikke er en DAG

**Q18**

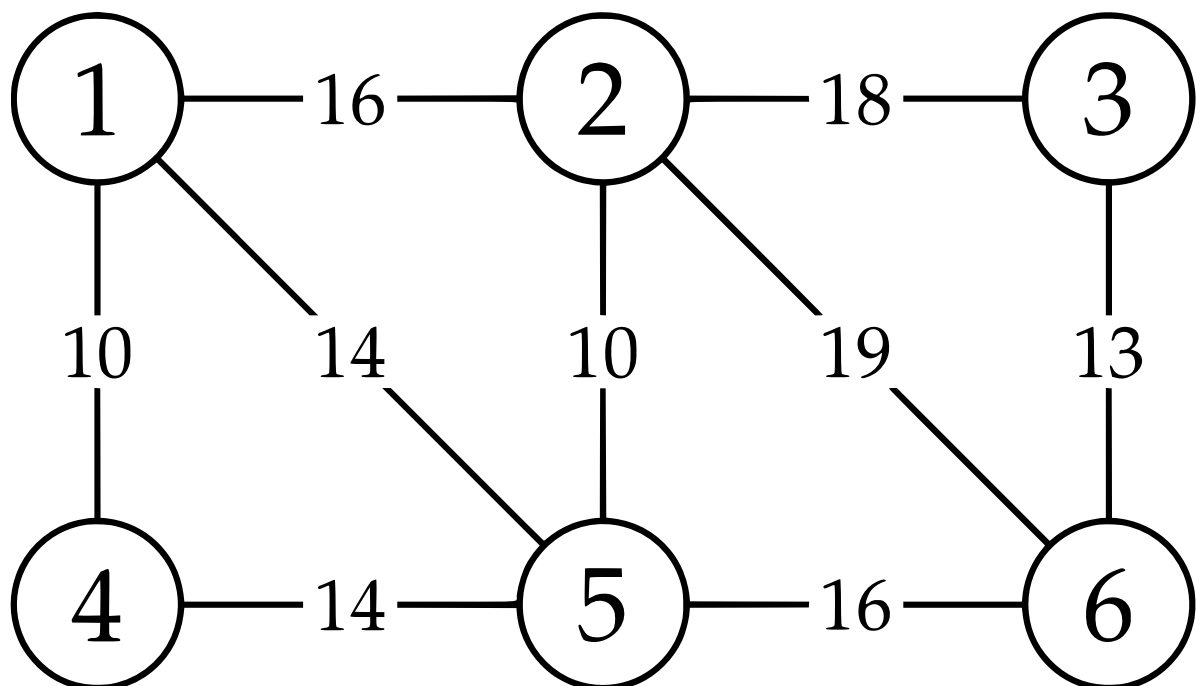
Hvilke påstander stemmer?

Svar:

- Prim er en grådig algoritme.
- Kruskal er en grådig algoritme.
- Kruskal velger alltid en lett kant som den neste kanten.

**Q19**

Hvis du utfører MSTSvar: Kruskal på grafen i denne oppgaven, hvilken kant vil velges som den femte i rekken? Det vil si, hvilken kant vil være den femte som legges til i løsningen? Oppgi kanten på formen (i, j), der  $i < j$ .



Svar: (5, 6)

**Q20**

Anta at du har korteste vei mellom nodene A og C, som går gjennom node B. Hvilken påstand er korrekt?

Svar: Vi har også korteste veg mellom A og B, og B og C

**Q21**

Hva er kjøretiden til BellmanSvar: Ford?

Svar:  $O(VE)$

**Q22**

Hvilke(n) påstand(er) er korrekt(e)?

Svar: Dijkstra er en grådig algoritme.

**Q23**

Hva er kjøretiden til FloydSvar: Warshall?

Svar:  $O(V^3)$

**Q24**

Johnsons bruker andre algoritmer som subrutiner. Hvilke?

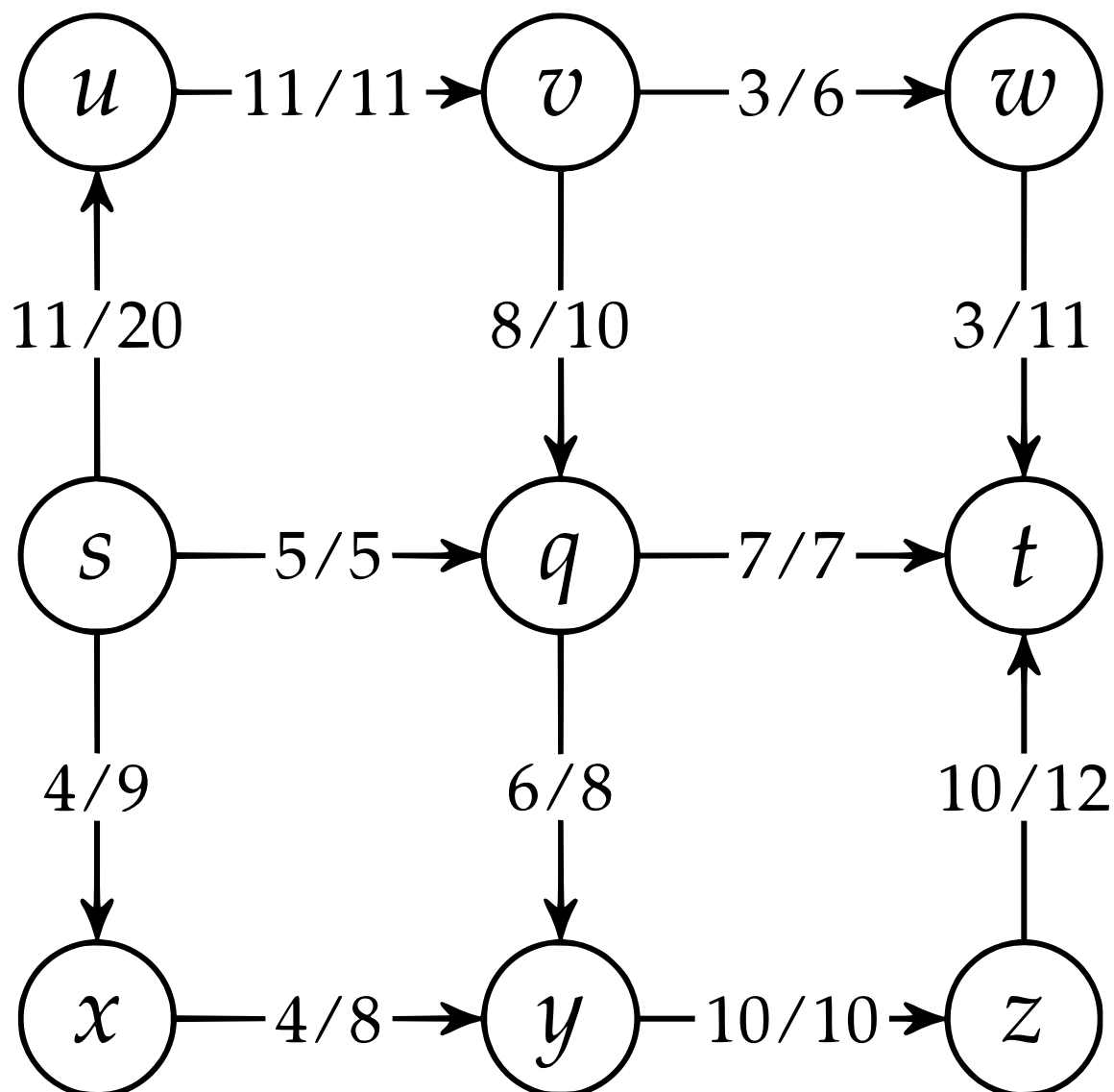
Svar: Dijkstra og BellmanSvar: Ford

**Q25**

Figuren i denne oppgaven viser flytnettverket  $G$ , med kilde  $s$ , sluk  $t$  og flyt  $f$ . Er flyten maksimal? Svar ja eller nei.

Hvis ja, oppgi også mengden av noder som kan nås (dvs., som det finnes stier til) fra  $s$  i  $G_f$ .

Hvis nei, oppgi også nodene i en flytforøkende sti (augmenting path), i rekkefølge.



Svar: Nei, flyten er ikke maksimal. En flytforøkende er  $[s, x, y, q, v, w, t]$ .

**Q26**

Hva er en matching?

Svar: En delmengde av alle kanter, der hver node er tilknyttet maks en kant fra delmengden.



**Q27**

Du vet at problem A er i NP og problem B er i NPC. Du vil vise at A også er i NPC. Hvilken vei vil du redusere?

Svar: Fra B til A.

**Q28**

Er følgende utsagn riktig?

«For ethvert språk L, hvis L er i NP, så er L ikke i coSvar: NP.»

Svar: Nei. For eksempel er ethvert problem i P både i NP og coSvar: NP.