

# TDT4120 Algoritmer og datastrukturer

Eksamen, 10. august 2023, 09:00–13:00

**Faglig kontakt** Magnus Lie Hetland  
**Hjelpemiddelkode** E

## Oppgaver

- 5% **1** Algoritme 1 sorterer tabellen  $A[1:n]$ . Hva er kjøretiden, som funksjon av  $n$ ?  
Oppgi svaret i asymptotisk notasjon.
- 5% **2** Hva er et spennetre?  
Det trenger ikke være minimalt.
- 5% **3** Hvilket problem løser FLOYD-WARSHALL?  
Her er vi ikke ute etter bare navnet på problemet, men en svært kort beskrivelse av hva problemet er.
- 5% **4** DIJKSTRA velger en node i hver iterasjon. Hvilken?
- 5% **5** Tellesortering (*counting sort*) har bedre kjøretid enn f.eks. flettesortering (*merge sort*). Hva er det vi krever av input til tellesortering som gjør dette mulig?
- 5% **6** Hva er konsekvensen av å finne en polynomisk algoritme for et problem i NPC?
- 5% **7** Dine venner Lurvik og Smartnes har laget hver sin sorteringsalgoritme, som rett og slett utfører to andre sorteringsalgoritmer etter hverandre:

```
LURVIK-SORT( $A, n$ )  
1 INSERTION-SORT( $A, n$ )  
2 MERGE-SORT( $A, 1, n$ )
```

```
SMARTNES-SORT( $A, n$ )  
1 MERGE-SORT( $A, 1, n$ )  
2 INSERTION-SORT( $A, n$ )
```

Hvilken av dem har best kjøretid i verste tilfelle? Forklar kort.

### Algoritme 1

```
FUNG-SORT( $A, n$ )  
1 for  $i = 1$  to  $n$   
2   for  $j = 1$  to  $n$   
3     if  $A[i] < A[j]$   
4       swap  $A[i]$  and  $A[j]$ 
```

- 5% **8** Lurvik og Smartnes skal på togferie. Det går direktetog mellom mange av byene de skal besøke, og Lurvik vil finne en rute som går innom hver by nøyaktig én gang, om mulig. Smartnes mener det er urealistisk. Hva mener du?

Det er her snakk om å lage en effektiv algoritme for å løse problemet generelt.

- 5% **9** Du skal finne et passord som består av  $n$  tegn fra et alfabet av størrelse  $k$ . Du prøver ett og ett passord (*brute force*). Hvor mange passord må du prøve før du finner det rette? Oppgi svaret i asymptotisk notasjon.

Du kan anta at du kjenner både  $n$  og  $k$ .

- 5% **10** Tre menn (Lurvik, Smartnes og Visdal) og tre kvinner (Gløgsund, Klokland og Flinckenhagen) har følgende preferanser:

|                |                                   |
|----------------|-----------------------------------|
| Lurvik:        | Gløgsund, Flinckenhagen, Klokland |
| Smartnes:      | Gløgsund, Klokland, Flinckenhagen |
| Visdal:        | Klokland, Flinckenhagen, Gløgsund |
| Gløgsund:      | Lurvik, Smartnes, Visdal          |
| Klokland:      | Visdal, Smartnes, Lurvik          |
| Flinckenhagen: | Lurvik, Smartnes, Visdal          |

Lurvik er matchet med Flinckenhagen, Smartnes er matchet med Gløgsund og Visdal er matchet med Klokland. Er matchingen stabil, eller finnes det et blokkerende par (*blocking pair*)? Hvem er det, i så fall? Forklar kort.

- 5% **11** Det følgende er hentet fra COUNTING-SORT:

```
11 for  $j = 1$  downto 1
12   ████████ =  $A[j]$ 
13    $C[A[j]] = C[A[j]] + 1$ 
```

Hva skal den sensurerte biten være?

- 5% **12** Løs følgende rekurrens:

$$T(n) = 2T(n/2) + n/\lg n$$

Oppgi svaret med asymptotisk notasjon.

- 5% **13** Tabellen  $A = \langle 9, 8, 5, 7, 1, 3, 2, 4, 6 \rangle$  representerer en haug. Hvordan ser tabellen ut etter første iterasjon av HEAPSORT?

Du skal altså utføre den første av  $n - 1$  iterasjoner. Svar ved å liste opp elementene i tabellen. Oppgi hele tabellen, inkludert deler som ikke lenger er en del av haugen.

- 5% **14** Flytnett (*flow networks*) kan defineres på litt forskjellige vis, men i versjonen i pensum tillates ikke antiparallele kanter (dvs., at man både har en kant fra  $u$  til  $v$  og en kant fra  $v$  til  $u$ ). Hvor stor begrensning er dette? Forklar kort.

- 5% **15** Algoritme 2 finner antall mulige permutasjoner av elementene i mengden  $S$ , rekursivt. Hva taler for og imot bruk av memoisering for å optimere den?

Du kan f.eks. bruke en hashtabell med mengder som nøkler.

## Algoritme 2

```
PERMUTATIONS(S)
1  if S == ∅
2    return 1
3  else n = 0
4    for each element x ∈ S
5      n = n + PERMUTATIONS(S - {x})
6    return n
```

- 5% **16** Beskriv hvordan du kan bruke rekursjon til å finne avstanden fra startnoden  $s$  til en gitt node  $v$  i en vektet, rettet graf.

**Merk:** Det er forventet at løsningen vil ha eksponentiell kjøretid.

Her kan du svare svært kort. Det kreves ingen grundig pseudokode e.l. Du kan anta at det finnes en sti fra  $s$  til enhver annen node i grafen.

- 5% **17** Et byggefirma har flere store oppdrag og skal fordele sine ansatte på disse. Hvert prosjekt har et sett med roller (tømrer, elektriker, rørlegger, etc.) og et antall som trengs av hver av disse. Hver ansatt er kompetent til å fylle én eller flere slike roller, men kan maksimalt delta i ett prosjekt, og fyller da nøyaktig én rolle. For å holde reiseavstandene nede kan hver ansatt bare bli tilordnet et prosjekt innenfor en gitt avstand fra hjemstedet.

Hvordan ville du ha funnet en gyldig fordeling?

- 5% **18** I beviset for at CIRCUIT-SAT er NP-komplett konstrueres en logisk krets som simulerer en datamaskin som utfører en verifikasjonsalgoritme. Hva er input for denne kretsen?

Her trenger du ikke beskrive konstante «inputs», bare dem man står fritt til å sette til 0 eller 1 for å løse oppfyllbarhetsproblemet. Du kan svare svært kort.

- 5% **19** Din venn Gløgsund har klart å slette alle mellomrom og all tegnsetting i en avhandling hun skriver på, og hun vil ha din hjelp til å splitte teksten opp i enkelt-ord. Det du har å hjelpe deg med er en liste med gyldige ord, og en oversikt over ord som aldri forekommer ved siden av hverandre. Beskriv en algoritme som løser problemet.

Det kan være flere gyldige løsninger. I så fall holder det at du finner én av dem.

- 5% **20** Anta at du har en prosedyre  $A$  som avgjør beslutningsproblemet VERTEX-COVER i konstant tid. Beskriv hvordan du kan bruke  $A$  til å finne et minst mulig nodedekke. Løsningen din skal ha så lav asymptotisk kjøretid som mulig. Gitt denne kjøretiden, skal den bruke så få kall til  $A$  som mulig. (Du skal altså ikke øke den asymptotiske kjøretiden bare for å redusere antall kall til  $A$ .)

Merk at du her faktisk skal finne nodedekket, ikke bare størrelsen.