Informasjon

Du måtte klare 9 av 17 oppgaver for å få øvingen godkjent.

Finner du feil, mangler eller forbedringer, ta gjerne kontakt!

Oppgave 2

«*Kjør algoritmen én gang fra hver node, totalt* $\Theta(\mathrm{V})$ *ganger.*» er riktig.

Kommentar:
Dette står nevnt på side 700 i læreboka. Merk at $\Theta(\mathrm{V})$ angir antall ganger Dijkstra må kjøres, ikke kjøretiden til en slik algoritme.

Oppgave 3

«Bellman-Ford, $O(\mathrm{V}^2\mathrm{E})$» er riktig.

Kommentar:
Siden det kun er oppgitt at det ikke finnes negative sykler (en nødvendighet for at problemet skal ha en løsning) og ikke spesifisert noe om negative kantvekter, kan vi ikke generelt bruke Dijkstra til å løse problemet. Dermed må vi bruke Bellman-Ford. Denne har kjent kjøretid $O(\mathrm{VE})$ (side 651 i læreboka), og total kjøretid for å kjøre den en gang fra hver node i grafen blir dermed $O(\mathrm{V}^2\mathrm{E})$.

Oppgave 4

«*Hvor man kom fra, på korteste vei fra* i *til* j» er riktig.

Kommentar:
Dette står i innledningen til kapittel 25, på side 685 i læreboka.

Oppgave 5

«*Enten er* $i = j$ *eller så er det ingen sti fra* i *til* j» er riktig.

Kommentar:
Dette står også på side 685 i læreboka.

Oppgave 6

«$O(V^3)$» er riktig.

Kommentar:
I læreboka står følgende om FLOYD-WARSHALL på side 694–695: «Its input is an $n \times n$ matrix W defined as in equations (25.1)», og «…the algorithm runs in time $\Theta(n^3)$». Matrisen W er en vektmatrise for grafen $G = (V, E)$, hvor

$$w_{ij} = \begin{cases} 0 & \text{om } i = j, \\ \text{vekten til den rettede kanten } (i,j) & \text{om } i \neq j \text{ og } (i,j) \in E, \\ \infty & \text{om } i \neq j \text{ og } (i,j) \notin E. \end{cases} \tag{25.1}$$

Fra dette er det tydelig at $n = |V|$, og kjøretiden blir dermed $\Theta(V^3)$. Fra $\Theta(V^3) = O(V^3)$ kan vi da løse oppgaven.

Oppgave 7

«$\Theta(V^2)$» er riktig.

Kommentar:
Lærebokas implementasjon lager $|V|$ ulike $|V| \times |V|$-matriser, og bruker dermed $\Theta(V^3)$ lagringsplass. En mer effektiv måte er å bare oppdatere den sammen matrisen underveis som følger:

```
FLOYD-WARSHALL'(W)
1  n = W.rows
2  D = W
3  for k = 1 to n
4      for i = 1 to n
5          for j = 1 to n
6              d_ij = min(d_ij, d_ik + d_kj)
7  return D
```

I oppgave **25.2-4** på side 699 i læreboka blir du bedt om å vise at denne implementasjonen er korrekt.

Oppgave 8

«*Kun utsagn 2 er sant.*» er riktig.

Kommentar:
Før den ytterste **for**-løkken i FLOYD-WARSHALL settes D lik W, noe som vil si at $d_{ii} = 0$ for $i = 1, 2, \ldots, |V|$. Siden vi i linje 7 d_{ii} lik minimum av sin nåværende verdi og summen av to andre tall, kan vi dermed aldri få d_{ii} lik et positivt tall. Utsagn 1 kan dermed ikke stemme.
Om det finnes en sti med negativ vekt fra node i tilbake til seg selv vil derimot d_{ii} til slutt bli

negativ. Vi kan med andre ord kontrollere diagonalen i D etter at FLOYD-WARSHALL er ferdig for å oppdage negative sykler i en graf.

Oppgave 9

```
def general_floyd_warshall(A, f, g):
    n = len(A)
    for k in range(n):
        for i in range(n):
            for j in range(n):
                A[i][j] = f(A[i][j], g(A[i][k], A[k][j]))
    return A
```

Oppgave 10

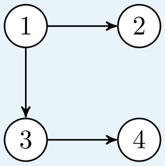
«$t_{ij}^{(k)} = t_{ij}^{(k-1)} \vee \left(t_{ik}^{(k-1)} \wedge t_{kj}^{(k-1)}\right)$» er riktig.

Kommentar:
Som i læreboka angir $\vee$ logisk OR, og $\wedge$ angir logisk AND. Se side 697 i læreboka for en utredning av uttrykket.

Oppgave 11

«*Ingen av de andre alternativene.*» er riktig.

Kommentar:
Fra definisjonen av $t_{ij}^{(k)}$ får vi ingen informasjon om lengden til en sti. At $t_{ij}^{k} = 0$ forteller oss at det ikke går noen sti fra node i til node j via nodene i mengden $\{1, 2, \ldots, k\}$.
I følgende graf går det en sti fra node 1 til node 4 med lengde 2, men per definisjon vil $t_{1,4}^{(2)} = 0$ ettersom $3 \notin \{1, 2\}$.



Oppgave 12

```
def transitive_closure(T, general_floyd_warshall):
    def f(x, y):
        return x or y

    def g(x, y):
        return x and y

    general_floyd_warshall(T, f, g)
```

Kommentar:I første del av koden omformer vi den gitte matrisen `D` til en binærmatrise. Resten av koden setter ganske enkelt `f = or` og `g = and`, slik at oppdateringsregelen blir som i svaret på forrige oppgave.

Oppgave 13

«Dijkstra *og* Bellman-Ford» er riktig.

Kommentar:
Se delkapittel 25.3 i læreboka.

Oppgave 14

«$O(VE \lg V)$» er riktig.

Kommentar:
Dette er oppgitt på side 704 i læreboka.

Oppgave 15

«*Revekting av kantvekter*» er riktig.

Kommentar:
Dette er diskutert i detalj i delkapittel 25.3 i læreboka.

Oppgave 16

«$O(V^3)$» er riktig.

Kommentar:
På både side 700 og side 704 i læreboka er det oppgitt at kjøretiden til JOHNSON er $O(V^2 \lg V + VE)$ når en bruker en Fibonacci-haug til å implementere minimumskøen i DIJKSTRA. Videre ser vi fra beskrivelsen at vi i en komplett digraf vil ha $|E| = |V|(|V| - 1) = \Theta(V^2)$. Setter vi dette inn i det oppgitte uttrykket for kjøretiden får vi da $O(V^2 \lg V + V^3) = O(V^3)$.

Oppgave 17

```python
import functools

def schulze_method(A):
    n = len(A)
    for i in range(n):
        for j in range(n):
            if A[i][j] > A[j][i]:
                    A[j][i] = 0
            elif A[i][j] < A[j][i]:
                    A[i][j] = 0
            else:
                    A[j][i] = 0
                    A[i][j] = 0

# Floyd-Warshall med max og min:
for k in range(n):
    for i in range(n):
        for j in range(n):
            A[i][j] = max(A[i][j], min(A[i][k], A[k][j]))

for i in range(n):
    A[i][i] = 0

result = list(range(n))
return sorted(
    result,
    key=functools.cmp_to_key(lambda x, y: A[x][y] - A[y][x]),
    reverse=True
)
```

Kommentar:
Denne koden bruker `functools.cmp_to_key` (en forkortelse for *comparison to key*) for å sortere listen over kandidater. Om du vil vite mer om denne funksjonaliteten, ta gjerne en titt i dokumentasjonen.

Oppgave 18

For eksamensoppgavene, se løsningsforslaget til den gitte eksamenen.

For oppgaver i CLRS, så finnes det mange ressurser på nettet som har fullstendige eller nesten fullstendige løsningsforslag på alle oppgavene i boken. Det er ikke garantert at disse er 100% korrekte, men de kan gi en god indikasjon på om du har fått til oppgaven. Det er selvfølgelig også mulig å spørre studassene om hjelp med disse oppgavene.

Et eksempel på et ganske greit løsningsforslag på CLRS, laget av en tidligere doktorgradsstudent ved Rutgers, finnes her.