INSTITUT FOR DATALOGI, AARHUS UNIVERSITET

| |
|---|
| Science and Technology |
| EKSAMEN |
| **Algoritmer og Datastrukturer 2 (2003-ordning)** |
| Antal sider i opgavesættet (incl. forsiden): 11 (elleve) |
| Eksamensdag: Fredag den 23. juni 2017, kl. 9.00-13.00 |
| Tilladte medbragte hjælpemidler: Alle sædvanlige hjælpemidler (lærebøger, notater, lommeregner). Computer må ikke medbringes. |
| Materiale der udleveres til eksaminanden: |

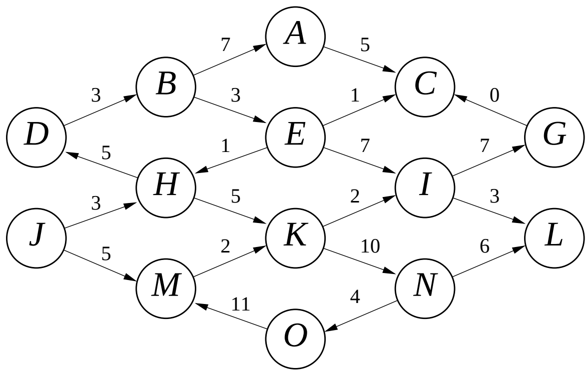
OPGAVETEKSTEN
BEGYNDER
PÅ NÆSTE SIDE

—oOo—

**Opgave 1** (25%)

**Bemærk:** *Bagerst i eksamenssættet findes sider til at angive svarene til opgave 1 på og som afleveres som del af eksamensbesvarelsen.*

Betragt nedenstående vægtede orienterede graf. Det antages, at grafen er givet ved *incidenslister*, hvor incidenslisterne er sorteret alfabetisk.

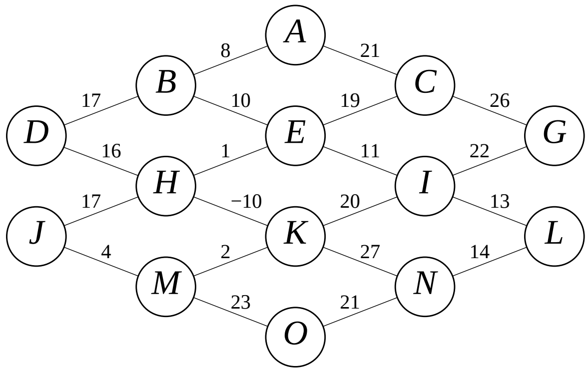



**Spørgsmål a:** Angiv et BFS-træ for ovenstående graf, hvor BFS-gennemløbet starter i **knuden** $J$. Angiv kanterne i BFS-træet, BFS-numrene for knuderne, og rækkefølgen knuderne bliver indsat i køen $Q$ i BFS-algoritmen. $\square$

**Spørgsmål b:** Angiv et DFS-træ for ovenstående graf, hvor DFS-gennemløbet starter i **knuden** $J$, og en DFS-nummerering af knuderne. Angiv for hver knude "discovery time" og "finishing time". Marker for hver kant om det er en "tree edge" (T), "back edge" (B), "cross edge" (C) eller "forward edge" (F). $\square$

**Spørgsmål c:** Angiv de stærke sammenhængskomponenter i ovenstående graf. For hver komponent angiv knuderne i komponenten. $\square$

**Spørgsmål d:** Angiv et korteste veje (SSSP) træ for ovenstående graf, når korteste veje beregningen sker med hensyn til **startknuden** $J$. For hver knude $v$ angiv også afstanden fra startknuden $J$ til $v$, og angiv rækkefølgen knuderne tages ud af prioritetskøen $Q$ i Dijkstra's algoritme. $\square$

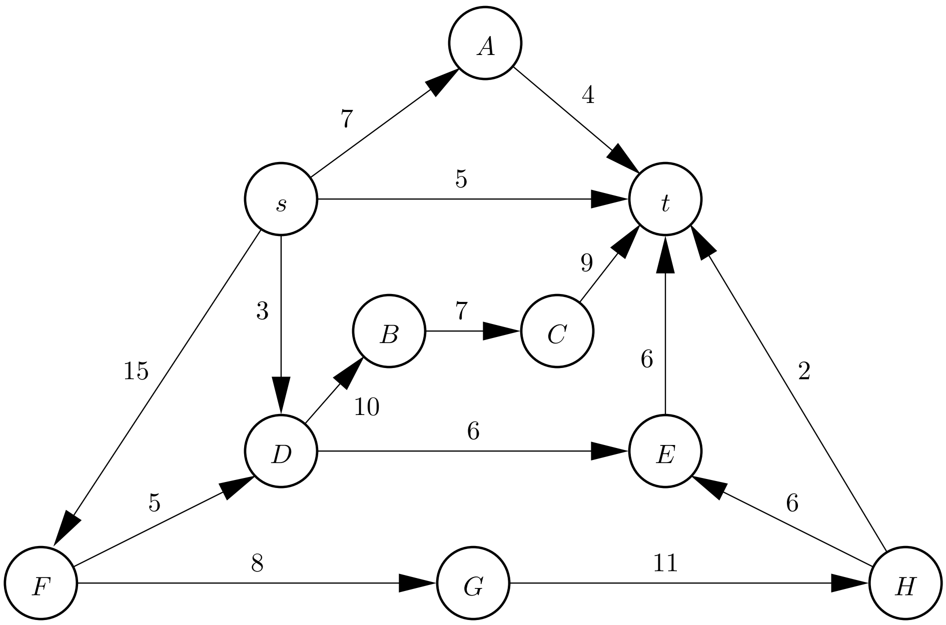



**Spørgsmål e:** Angiv kanterne i et minimum udspændende træ (MST) for ovenstående uorienterede graf med vægte på kanterne. Angiv i hvilken rækkefølge knuderne bliver taget ud af prioritetskøen $Q$ i Prim's algoritme, når denne starter i **knuden** $A$. $\square$

**Opgave 2** (15%)

**Bemærk:** *Bagerst i eksamenssættet findes en side til at angive svarene til opgave 2 på og som afleveres som del af eksamensbesvarelsen.*

Betragt nedenstående netværk med de angivne kapaciteter på kanterne.




**Spørgsmål a:** Angiv en maksimal strømning (maximum flow) fra $s$ til $t$ i netværket (angiv for hver kant strømningen langs kanten), angiv værdien af en maksimal strømning, og angiv et snit (cut) (dvs. opdeling af knuderne i to disjunkte mængder) hvor kapaciteten af snittet er lig værdien af en maksimal strømning. □

**Spørgsmål b:** Betragt Edmonds-Karp algoritmen anvendt på ovenstående graf til beregning af en maksimal strømning. Angiv de forbedrende stier (augmenting paths) der anvendes under udførslen af Edmonds-Karp algoritmen. For hver forbedrende sti angiv knuderne på stien og strømningen, man forbedrer med, langs stien. □

**Opgave 3** (20%)

I denne opgave antages at vi har givet en orienteret graf $G = (V, E)$, med $n = |V|$ knuder og $m = |E|$ kanter. For to knuder $u$ og $v$ definerer vi afstanden $\text{dist}(u, v)$ som længden af den korteste orienterede sti fra $u$ til $v$, hvor længden af en sti er antallet af kanter på stien. Hvis $v$ ikke kan nås fra $u$ så er $\text{dist}(u, v) = +\infty$.

**Spørgsmål a:** Givet en knude $u$, beskriv en algoritme der finder en knude $v$, som er længst væk fra $u$, dvs. $\text{dist}(u, v) = \max_{w \in V} \text{dist}(u, w)$ ($+\infty$, hvis der findes en knude $v$ der ikke kan nås fra $u$). Angiv algoritmens udførselstid. □

Vi kalder en knude $u$ en *broadcast* knude, hvis der for alle knuder $v \in V$ findes en sti fra $u$ til $v$. *Broadcast radius* $\text{rad}(u)$ for $u$ er den maksimale afstand til en anden knude, dvs. $\text{rad}(u) = \max_{v \in V} \text{dist}(u, v)$. En knude med minimal broadcast radius kaldes for en *central broadcast knude*, og vi lader grafens broadcast radius være $\text{rad}(G) = \min_{u \in V} \text{rad}(u)$.

**Spørgsmål b:** Beskriv en algoritme der finder en central broadcast knude i en orienteret graf $G$, under antagelse at der findes mindst en broadcast knude i $G$. Angiv algoritmens udførselstid. En ideel besvarelse opnår udførselstid $O(n^3)$. □

**Spørgsmål c:** Beskriv en algoritme, der afgør om en orienteret graf $G$ har en broadcast knude. Angiv algoritmens udførselstid. En ideel besvarelse opnår udførselstid $O(n+m)$. □

Vi ønsker nu at anvende to broadcast knuder, som broadcaster til hver sin del af grafen $G$, således at den maximale broadcast radius er mindst mulig. Det antages at $G$ har mindst en broadcast knude. Formelt ønsker vi at finde en disjunkt opdeling af $V$ i to delmængder $V_1$ og $V_2$, således at $\max(\text{rad}(G_1), \text{rad}(G_2))$ er mindst mulig, hvor $G_1 = (V_1, E_1)$ og $G_2 = (V_2, E_2)$ og $E_i$ er præcis de kanter i $E$ som forbinder knuder i $V_i$, dvs. $E_i = E \cap (V_i \times V_i)$.

**Spørgsmål d:** Beskriv en algoritme, der finder den minimale broadcast radius, når der må anvendes to broadcast knuder. Angiv algoritmens udførselstid. □

**Opgave 4** (20%)

En sekvens af reelle tal $y_1 < y_2 < \cdots < y_m$ siges at være *accelererende* hvis $y_{i+1} - y_i > y_i - y_{i-1}$, for alle $1 < i < m$. F.eks. er $2, 4, 7, 11$ en accelererende sekvens da $11 - 7 > 7 - 4 > 4 - 2$, hvorimod $3, 6, 8$ ikke er en accelererende delsekvens da $8 - 6 < 6 - 3$.

For en sekvens af $n$ reelle tal $x_1 < x_2 < \cdots < x_n$, ønsker vi at finde en længste accelererende delsekvens. F.eks. er $2, 4, 8$ en længste accelererende delsekvens af $2, 4, 6, 8, 10$.

**Spørgsmål a:** Angiv en længste accelererende delsekvens af sekvensen

$$2, 6, 7, 9, 12, 16, 18, 25, 33$$

□

For $1 \leq i < j \leq n$ lader vi $A(i, j)$ betegne længden af en længste accelererende delsekvens af $x_1, \ldots, x_j$, hvor $x_i$ og $x_j$ er de sidste to elementer i delsekvensen. For $1 \leq i \leq n$ definerer vi $A(i, i) = 1$.

$A(i, j)$ kan bestemmes ved følgende rekursionsformel, hvor $1 \leq i \leq j \leq n$.

$$A(i,j) = \begin{cases} 1 & \text{hvis } i = j \\ 1 + \max\{A(k,i) \mid 1 \leq k \leq i \land x_j - x_i > x_i - x_k\} & \text{hvis } i < j \end{cases}$$

**Spørgsmål b:** Udfyld nedenstående tabel for $A(i, j)$ for sekvensen $3, 9, 11, 13, 16$.

| $A(i,j)$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |

□

**Spørgsmål c:** Angiv en algoritme baseret på dynamisk programmering, der givet en sekvens af $n$ reelle tal $x_1 < \cdots < x_n$, beregner længden af en længste accelererende delsekvens. Angiv algoritmens udførselstid. □

**Spørgsmål d:** Udvid algoritmen til at rapportere en længste accelererende delsekvens af $x_1 < \cdots < x_n$. Angiv algoritmens udførselstid. □

**Opgave 5** (20%)

Givet en streng $T$ af længde $n$, ønsker vi at finde en delstreng $S$ af $T$, således at der findes to ikke-overlappende forekomster af $S$ i $T$. I det følgende antages at disse forekomster er på position $i$ og $j$, hvor $i < j$ og $j - i \geq |S|$. F.eks. for strengen $T$ = `bbabcabcabaa` forekommer delstrengen $S$ = `abc` på position $i = 3$ og $j = 6$. Bemærk at strengen $S'$ = `abcab` også forekommer to gange på positionerne 3 og 6, men at disse to forekomster overlapper og derfor ikke vil være et lovligt svar. Bemærk for den længere streng $T'$ = `bbabcabcabaabcabc` findes delstrengen $S'$ = `abcab` på positionerne 3, 6 og 12, hvor forekomsterne på position 3 og 12 er ikke-overlappende og derfor et lovligt svar.

I det følgende kan det antages at strengene er over et alfabet med $O(1)$ tegn og at et suffikstræ for en streng af længde $n$ over et alfabet med $O(1)$ tegn kan konstrueres i $O(n)$ tid.

**Spørgsmål a:** For nedenstående streng $T$, angiv en længste delstreng $S$ af $T$ og positioner $i$ og $j$, hvor der er to ikke-overlappende forekomster af $S$ i $T$ på positionerne $i$ og $j$ og $j - i \geq |S|$.

$T$ = `abcababcbbcababcbbcc`

□

**Spørgsmål b:** Beskriv en algoritme, der givet en streng $T$ af længde $n$, finder en længste delstreng $S$, således at der findes to ikke-overlappende forekomster af $S$ i $T$ på positioner $i$ og $j$, hvor $j - i \geq |S|$. Angiv algoritmens udførselstid. En ideel besvarelse opnår udførselstid $O(n)$. □

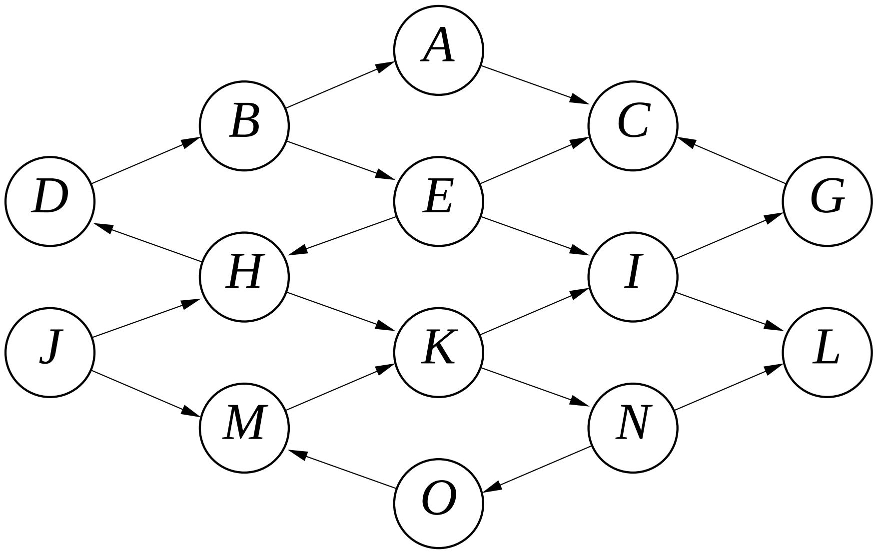
**Spørgsmål c:** Beskriv en algoritme, der givet en streng $T$ af længde $n$, finder en længste delstreng $S$, således at der findes *tre* ikke-overlappende forekomster af $S$ i $T$ på positioner $i$, $j$ og $k$, hvor $k - j \geq |S|$ og $j - i \geq |S|$. Angiv algoritmens udførselstid. En ideel besvarelse opnår udførselstid $O(n^2)$ eller bedre. □

| Årskort: | Navn: |
|---|---|



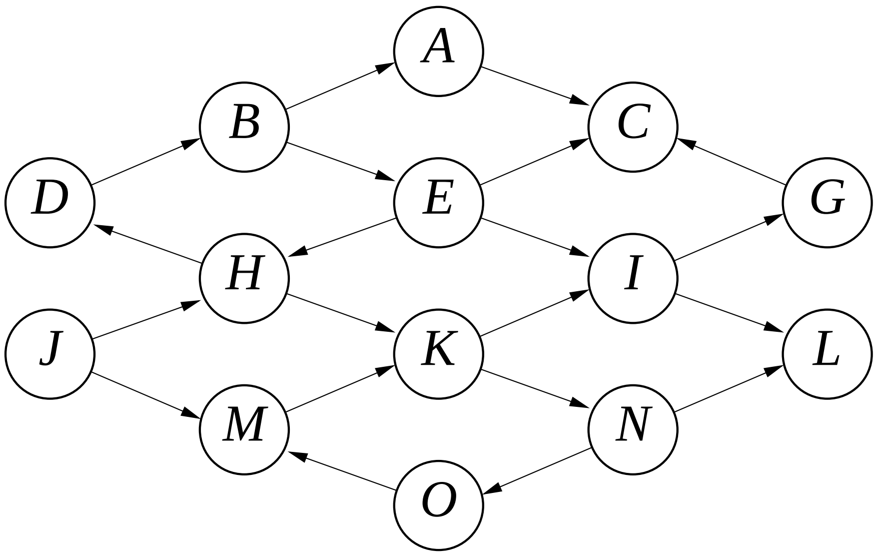
## Opgave 1 — Svar

### Spørgsmål a: BFS




Indsættelser i $Q$: ______________________________

### Spørgsmål b: DFS



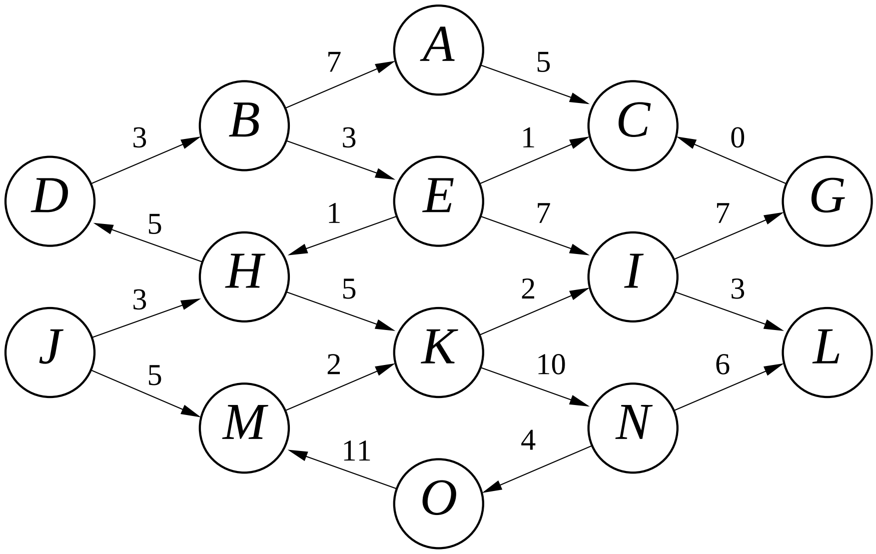
( blank side )

## Opgave 1 — Svar

### Spørgsmål c:

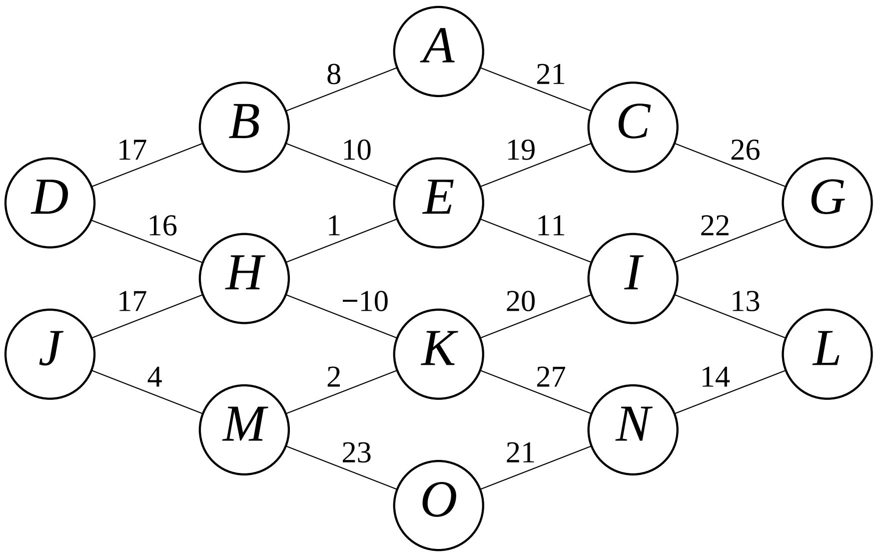
Stærke sammenhængskomponenter: ______________________

### Spørgsmål d: SSSP




Udtagelse fra $Q$: ______________________

### Spørgsmål e: MST

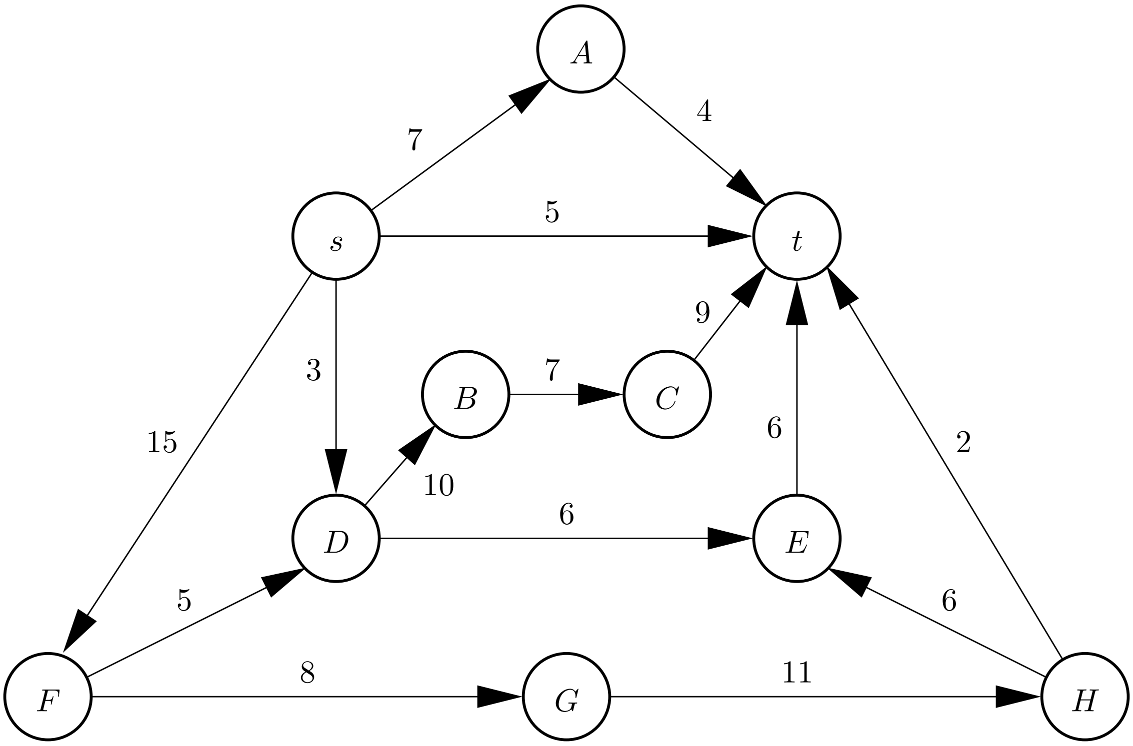



Udtagelse fra $Q$: ______________________

( blank side )

## Opgave 2 — Svar

**Spørgsmål a:**




Værdien af strømning: ______________________

Minimal snit: ______________________

**Spørgsmål b:**

| Forbedring | Forbedrende sti |
|---|---|
| | |