DANMARKS TEKNISKE UNIVERSITET

Skriftlig prøve, d. 17. maj 2022

Kursusnavn: Algoritmer og datastrukturer

Kursusnummer: 02326

Hjælpemidler: Skriftlige hjælpemidler er tilladt. Lommeregner er ikke tilladt. Computer er ikke tilladt.

Varighed: 4 timer

Cirka vægtning:

Opgave 1: 25%, Opgave 2: 20%, Opgave 3: 20%,

Opgave 4: 10%, Opgave 5: 15%.

Vægtningen er kun vejledende da der foretages en afsluttende heldhedsvurdering, som også medindregner opgaver fra CodeJudge (cirka 10%, helhedsvurdering).

Alle opgaver besvares ved at udfylde eller afkrydse de indrettede svarfelter, svarlinier og svarbokse nedenfor. Som opgavebesvarelse afleveres blot denne og de efterfølgende sider i udfyldt stand. I tilfælde af pladsmangel kan man eventuelt vedlægge ekstra sider.

Medmindre andet er angivet er basen på alle logaritmer 2.

Skriv dit fulde navn og studienummer.

Navn: ______________________________

Studienummer: ______________________________

1 Kompleksitet

1.1 (5%) Angiv for hver af nedenstående udsagn om de er korrekte:

	Ja	Nej
$6n + \frac{7n}{100} + \frac{113n}{27} = \Theta(n)$	☐	☐
$n^2(n^2 + 8n^3) = O(n^5 \log n)$	☐	☐
$2^n \cdot 2^n = O(2^{n+3})$	☐	☐
$\log n + (\log n)^2 + (\log n)^3 = O(\log n)$	☐	☐
$25\sqrt{n} + 0.32(\log n)^2 = \Omega((\log n)^3)$	☐	☐

1.2 (5%) Arranger følgende funktioner i voksende rækkefølge efter asymptotisk vækst.

Dvs. hvis funktionen $g(n)$ følger umiddelbart efter funktionen $f(n)$ i din liste, skal der gælde at $f(n) = O(g(n))$.

Angiv funktionens nummer i rækkefølgen (1,2,3,4,5) på linien.

$0.00231 \cdot 4^n$ __________

$33 \cdot (\log n)^2$ __________

$12 \cdot 2^n \cdot \log n$ __________

$\log n + (\log n)^2 + (\log n)^3$ __________

$7 \cdot \sqrt{n} + 5^2 \cdot \log(\sqrt{n})$ __________

1.3 (5%) Angiv køretiden af nedenstående algoritme.

Skriv dit svar i Θ-notation som funktion af n.

```
ALG1(n)
s = 0
for i = 1 to (2 · ⌈√n⌉) do
    for j = i to ⌈√n⌉ do
        s = s + i + j
    end for
end for
```

Tid: ______________________________

1.4 (5%) Angiv køretiden af nedenstående algoritme.

Skriv dit svar i Θ-notation som funktion af n.

```
ALG2(n)
c = n
j = c
while j ≥ log n do
    j = j − 1
end while
```

Tid: ______________________________

1.5 (5%) Angiv køretiden af nedenstående algoritme.

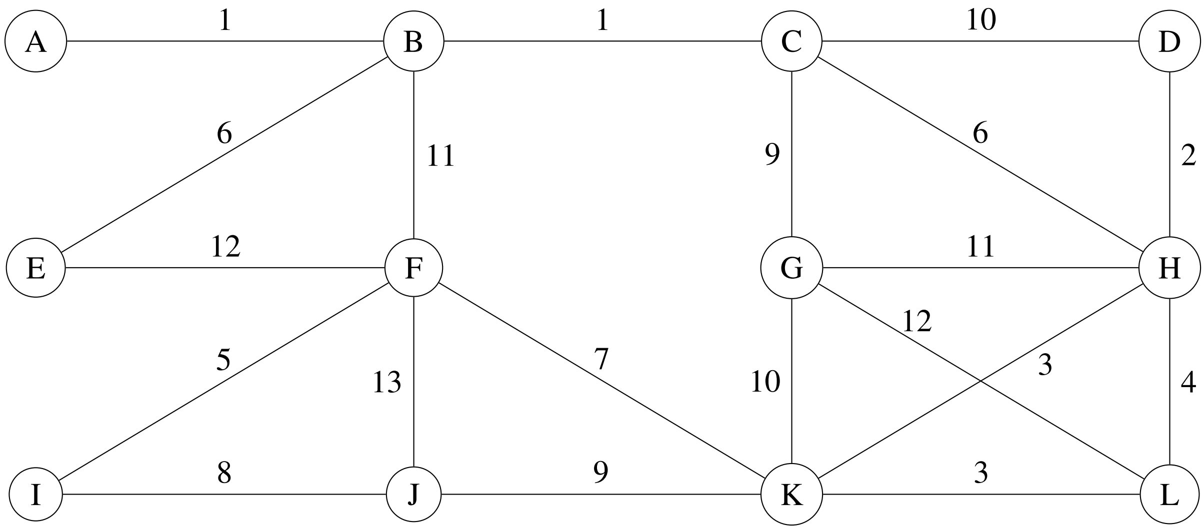
Skriv dit svar i Θ-notation som funktion af n.

```
ALG3(n)
if n ≤ 1 then
    return 1
else
    return ALG3(⌈n/2⌉)
end if
```

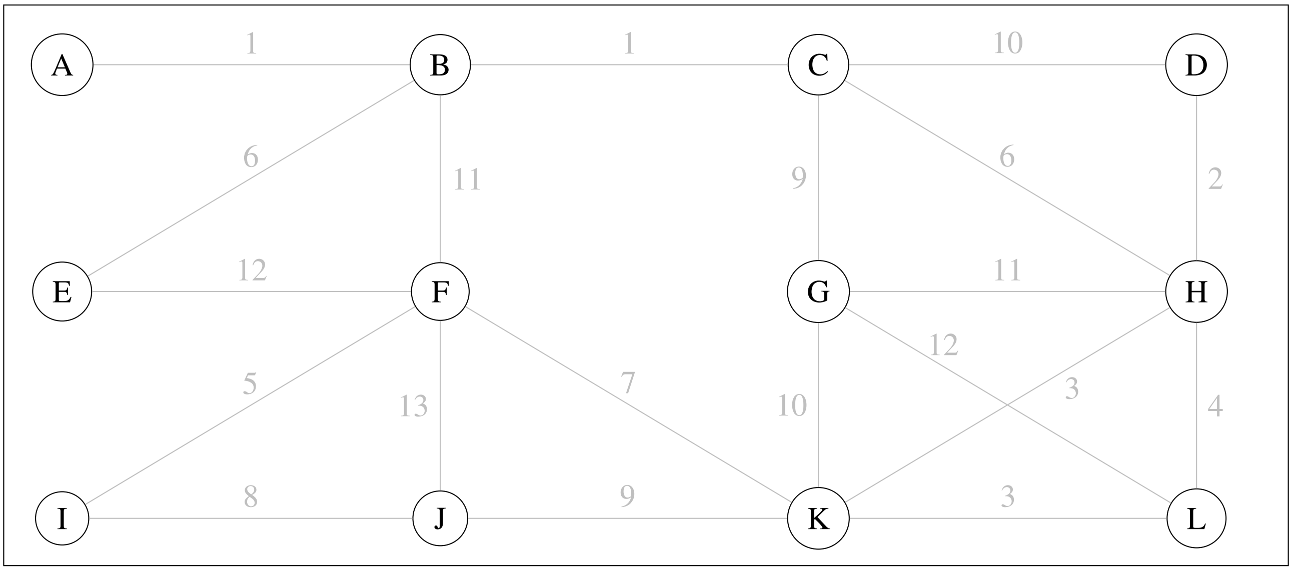
Tid: ______________________________

2 Grafer

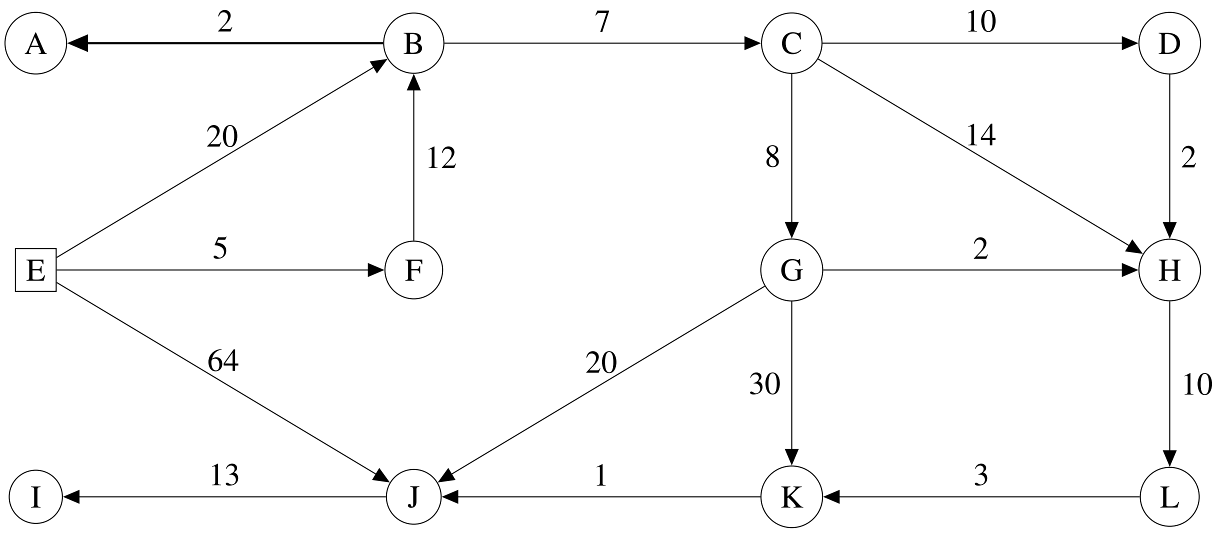
2.1 (5%) Betragt følgende graf:



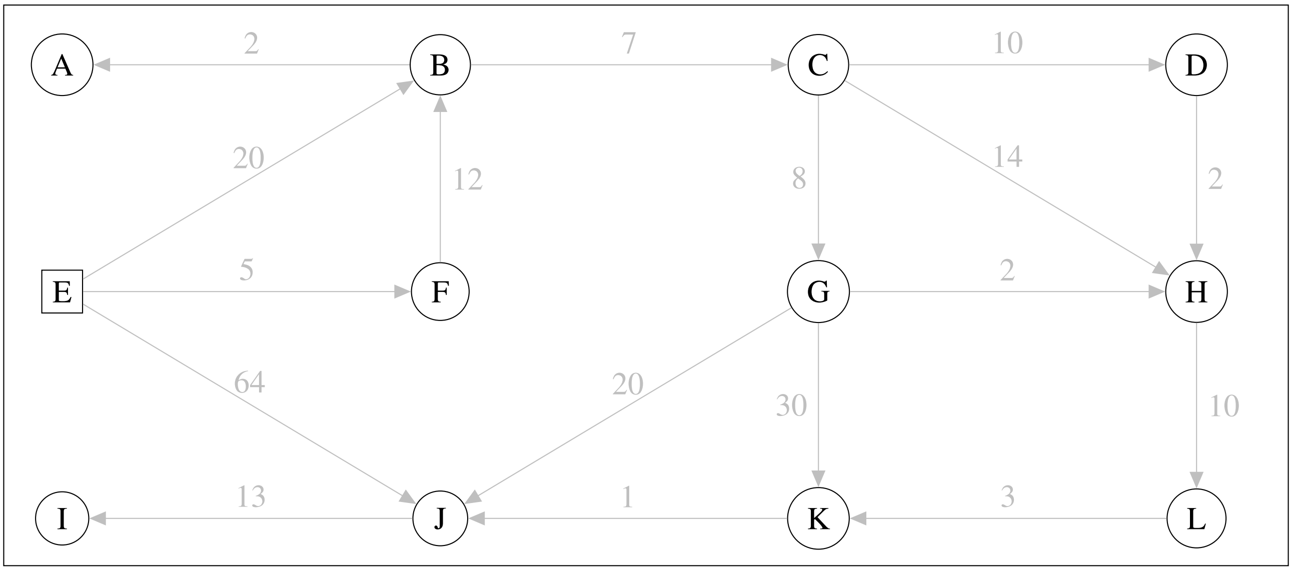
Angiv et mindste udspændende træ nedenfor, ved at markere de kanter, som er med i dit mindste udspændende træ.



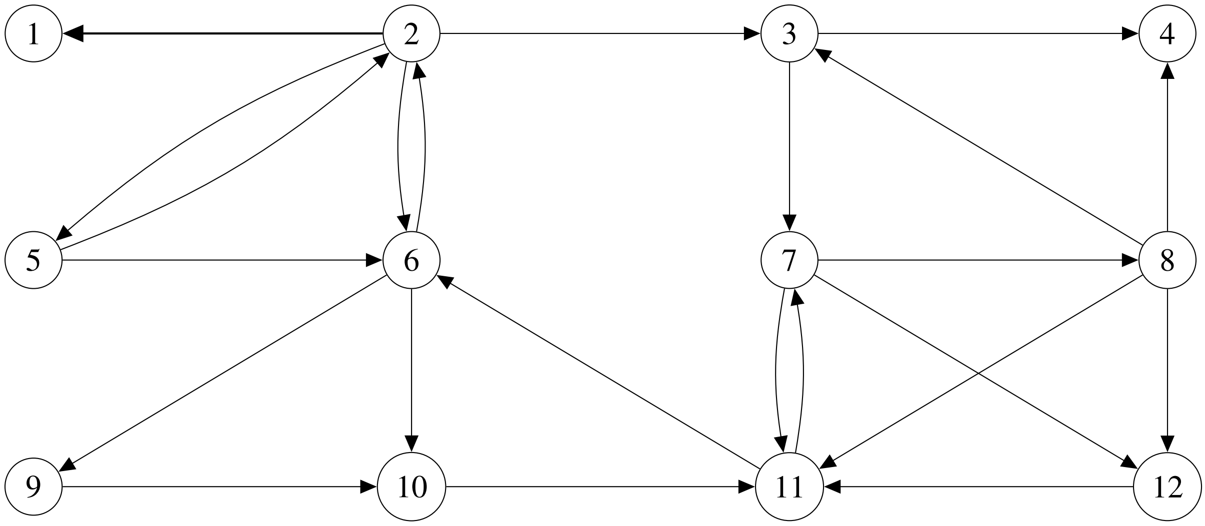
2.2 (5%) Betragt følgende graf:



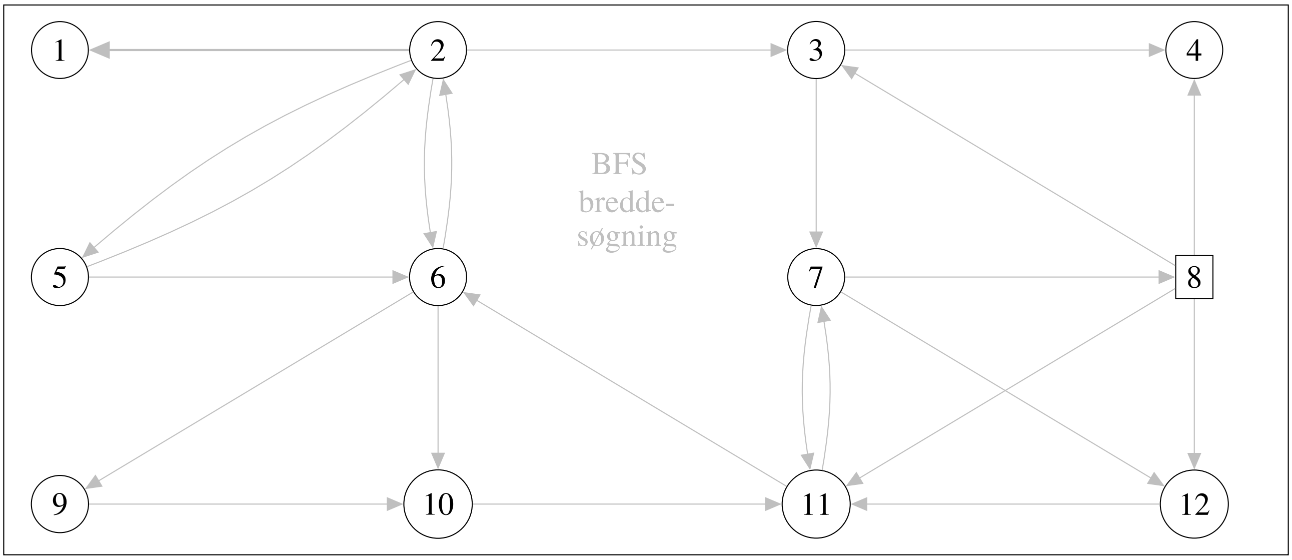
Angiv et korteste-veje-træ fra knuden E nedenfor, ved at markere de kanter, som er med i dit korteste-veje-træ.



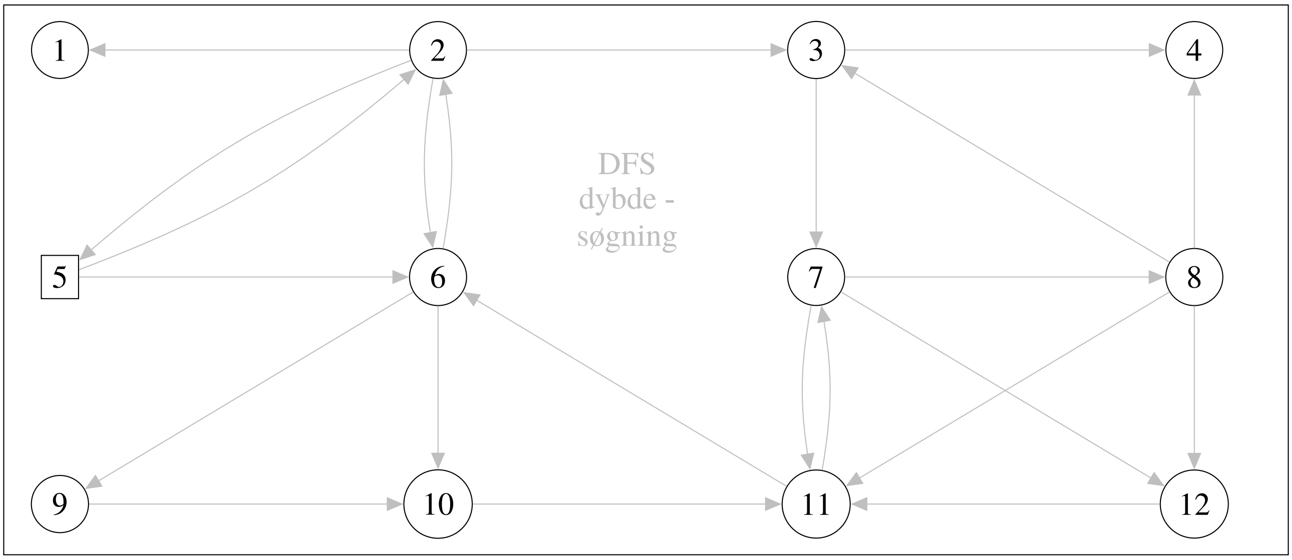
2.3 (10%) Betragt følgende graf:



Kør en bredde-først-søgning fra knuden 8 nedenfor, og marker de kanter, som er med i bredde-først-søgningstræet. Antag, at incidenslisterne er **sorterede**.



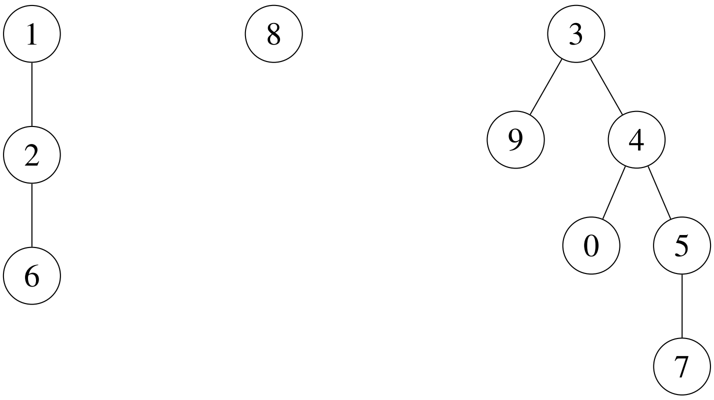
Kør en dybde-først-søgning fra knuden 5 nedenfor, og marker de kanter, som er med i dybde-først-søgningstræet. Antag, at incidenslisterne er **sorterede**.



3 Datastrukturer

3.1 (10%) Foren og find

Betragt følgende skov af træer, som repræsenterer en familie af mængder i en foren-og-find datastruktur, således som de bliver dannet af “hurtig forening”-algoritmen (quick union).



Angiv resultatet af følgende kald til funktionen Find:

Find(0) ____________________

Find(2) ____________________

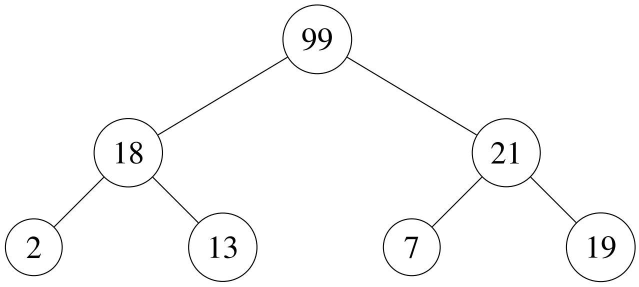
Find(6) ____________________

Find(7) ____________________

Find(8) ____________________

Stikompression. Tegn skoven, som den ser ud efter et kald til Find(5) med stikompression.

3.2 (5%) Betragt følgende binære max-hob:



Antag, vi udfører operationen `Extract-max`, og at vi efterfølgende indsætter tallet 20.

Tegn den resulterende hob

og skriv den på arrayform: ______________________________

3.3 (5%) Sammenligning af datastrukturer.

Antag vi ønsker at find max-min-summen af en talmængde, det vil sige, summen af det højeste og det laveste tal i mængden.

Eksempel: hvis vores mængde er tallene $\{9, 3, 1, 7\}$, så er max = 9 og min = 1, så derfor er max-min-summen $9 + 1 = 10$.

For hver af følgende datastrukturer, angiv, hvor hurtigt max-min-summen kan udregnes.

Bemærk at datastrukturen skal bruges 'som den er' – opgaven går ikke ud på at forbedre datastrukturen eller kombinere datastrukturer.

Angiv svaret i Θ-notation som funktion af n, som er talmængdens størrelse.

Hægtet liste ______________________________

Sorteret array ______________________________

Max-hob ______________________________

Min-hob ______________________________

Binært søgetræ ______________________________

4 Den nye cafe

Der er åbnet en ny cafe i nærheden af Algo University.

En studerende kommer forbi ved et tilfælde og opdager, at stedet både er godt og hyggeligt og tilbyder studenterrabat. Næste dag fortæller hun sine venner om stedet, og næste fyraften tager de alle på den nye cafe. Dagen efter fortæller de deres venner om stedet (de anbefaler det ikke før de har været der selv), og ved fyraften tager de alle også med på cafe. Sådan fortsætter det.

Du har fuldstændig viden om alle venskaber blandt Algo's universitetsstuderende, du kender den første studerende til at opdage cafeen, og du vil gerne forudsige den dag hvor der sker den største tilkomst af nye kunder i cafeen.

Input:

- Der er N studerende, og
- M venskaber mellem studerende,
- Studerende nr. s er den første til at opdage cafeen på dag 0.

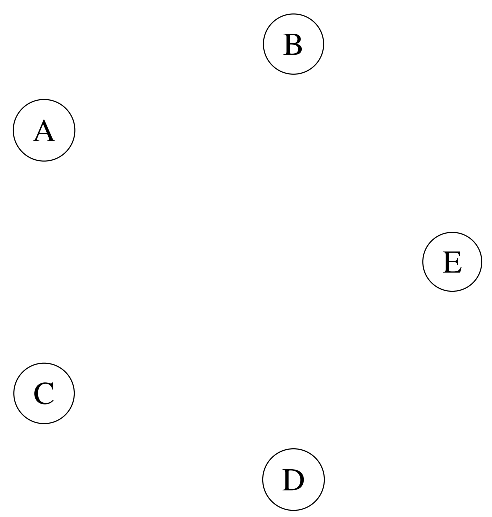
Alle venskaber er gengældte.

Output:

- Antallet af nye kunder på den dag, der kommer flest nye kunder,
- Nummerangivelse på den dag, der kommer flest nye kunder.

Eksempel: Studerende: Anna, Bente, Carlo, Denny og Edward. Venskaber: Anna er venner med Bente. Bente er venner med Carlo, Denny og Edward. Denny er venner med Edward. På første dag fortæller Anne til Bente om cafeen. På anden dag fortæller Bente til Carlo, Denny og Edward om cafeen. Output: Der kommer 3 nye kunder på dag nr. 2, og dette er det højeste antal.

4.1 (3%) Tegn venskaberne mellem Anna, Bente, Carlo, Denny og Edward som en graf.



4.2 (7%) Giv en algoritme til at finde dagen med den største tilkomst af nye gæster, og rapportere antallet af nye gæster. Argumenter for korrekthed, og analyser din algoritme som funktion af N og M.

5 Skisportsstedet

Vi vil gerne planlægge en tur ned ad bjerget på et skisportssted. På skisportstedet er der:

- x positioner, $p_1, p_2, \ldots, p_x$, og
- y løjper $s_1, s_2, \ldots, s_y$.
- z lifte $s_{y+1}, s_{y+2}, \ldots, s_{y+z}$.

Hver løjpe og hver lift s_i er defineret ved følgende:

- En startposition $\text{start}(s_i)$,
- En slutposition $\text{slut}(s_i)$,
- En varighed $\text{tid}(s_i)$, som er den tid, det tager at stå ski ned ad løjpen, eller den tid, det tager at køre op med liften. (Antal minutter.)

En tur fra position a til position b er en sekvens af løjper og/eller lifte, som starter i a, hvor hver efterfølgende løjpe eller lift starter hvor den foregående slutter, og hvor den sidste slutter i b.

5.1 (3%) Beskriv kort, hvordan man modellerer skisportstedet som en graf.

5.2 (6%) Vi er interesseret i at beregne, om vi kan komme fra bjergtoppen p_1 ned til positionen p_x, inden skisportsstedet lukker.

Giv en algoritme, der givet en beskrivelse af et skisportssted og en tidstærskel t afgør, om det er muligt at komme fra p_1 til p_x på højest t tid.

Argumenter for korrekthed, og analyser køretiden som funktion af parametrene: x, y, z.

5.3 (6%) Vi udvidder nu beskrivelsen af løjper og lifte, således at hver løjpe eller lift s_i også har:

- En sværhedsgrad level(s_i) som indikerer, hvor svær løjpen eller liften er at bruge. level(s_i) er et tal mellem 1 og x, hvor 1 er nemmest og x er sværest.

Vi ønsker nu en algoritme til at finde den **nemmeste** vej, indenfor en tidsgrænse.

Giv en algoritme, der givet beskrivelsen af et skisportssted og en tidsgrænse t finder den laveste sværhedsgrad g således at man kan komme fra p_1 til p_x indenfor tiden t uden at benytte løjper eller lifte med sværhedsgrad over g.

Argumenter for korrekthed, og analyser køretiden som funktion af parametrene: x, y, z.