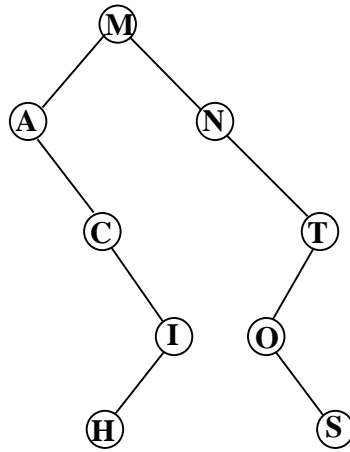


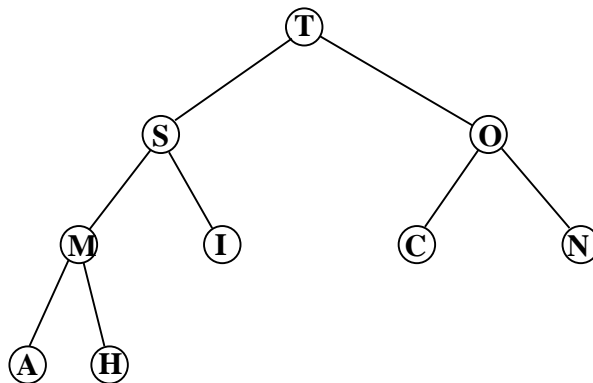
Vejledende løsninger til opgave 1, 2 og 3

Opgave 1: Træer

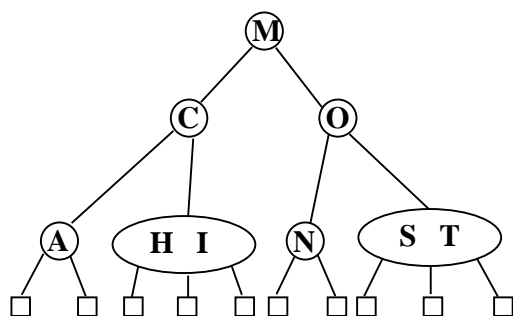
Spørgsmål 1.1



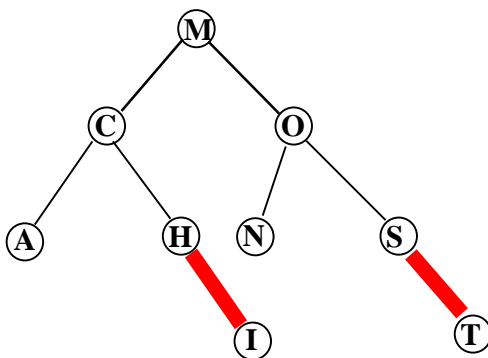
Spørgsmål 1.2



Spørgsmål 1.3

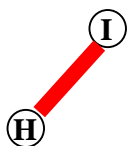


Spørgsmål 1.4

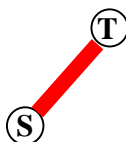


De røde grene er markeret med **fed** streg, de sorte med tynd streg.

HI-grenen og ST-grenen kan erstattes med henholdsvis

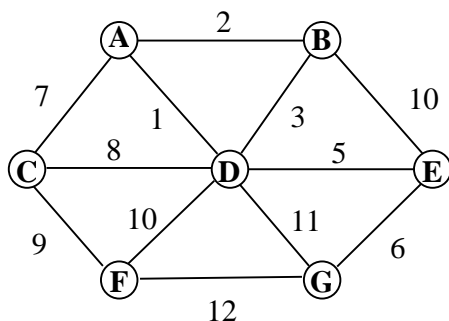


og

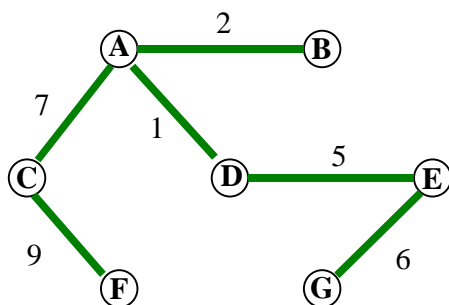


Opgave 2: Grafer

Spørgsmål 2.1



Spørgsmål 2.2



Summen af vægte er **30**.

Spørgsmål 2.3

	A	B	C	D	E	F	G
A	0	2	7	1	6	11	12
B	2	0	9	3	8	13	14
C	7	9	0	8	13	9	19
D	1	3	8	0	5	10	11
E	6	8	13	5	0	15	6
F	11	13	9	10	15	0	12
G	12	14	19	11	6	12	0

Svaret kan opnås ved inspektion af tegningen af grafen - eller, mere systematisk, ved hjælp af Floyds algoritme.

Opgave 3: Rekursion

Spørgsmål 3.1

1 3 4
1 2 3
1 2 4
2 3 4

Begrundelse (kræves ikke i opgaven):

For at få overblik over de rekursive metodekald tegnes stakken. Efter de tre første kald ser den således ud:

up	b = 5, c = 1
down	b = 2, c = 2, i = 4
up	b = 1, c = 3, i = 1

Hvert stakelement indeholder parametrene (b og c) og den lokale variabel (i). Til venstre for stakken er anført navnet på den tilhørende metode.

Det sidste metodekald har ingen effekt (for-løkken er "tom"). Nu afstakkes det øverste element, og i-værdien for det underliggende down-kald sænkes med 1, hvorefter up kaldes. Stakken ser nu således ud:

up	b = 4, c = 1
down	b = 2, c = 2, i = 3
up	b = 1, c = 3, i = 1

Herefter udføres kaldet `down(4, 0)`, og stakken får følgende udseende:

down	b = 5, c = 0
up	b = 4, c = 1, i = 4
down	b = 2, c = 2, i = 3
up	b = 1, c = 3, i = 1

Det bevirker, at metoden `print` kaldes. Eftersom i-værdierne bliver lagret fortløbende i arrayet `a`, udskrives `1 3 4` (efterfulgt af et linjeskift).

De to øverste elementer afstakkes (idet de respektive kald afsluttes), og i-værdien for det underliggende down-kald sænkes med 1.

Herefter kaldes `up`. Stakken ser nu således ud:

up	b = 3, c = 1
down	b = 2, c = 2, i = 2
up	b = 1, c = 3, i = 1

Nu udføres kaldet $\text{down}(4, 0)$.

down	$b = 4, c = 0$
up	$b = 3, c = 1, i = 3$
down	$b = 2, c = 2, i = 2$
up	$b = 1, c = 3, i = 1$

Dette bevirker, at arrayets indhold udskrives, d.v.s. 1 2 3 (efterfulgt af et linjeskift). Det øverste stakelement afstakkes, i -værdien for up -kaldet øges med 1, og $\text{down}(5, 0)$.

down	$b = 5, c = 0$
up	$b = 3, c = 1, i = 4$
down	$b = 2, c = 2, i = 2$
up	$b = 1, c = 3, i = 1$

Derfor udskrives 1 2 4 (efterfulgt af et linjeskift).

De tre øverste stakelementer afstakkes, og i -værdien for det underliggende up -kald øges med 1.

up	$b = 1, c = 3, i = 2$
----	-----------------------

Så udføres $\text{down}(3, 2)$, $\text{up}(4, 1)$, efterfulgt af $\text{down}(5, 0)$.

down	$b = 5, c = 0$
up	$b = 4, c = 1, i = 4$
down	$b = 3, c = 2, i = 3$
up	$b = 1, c = 3, i = 2$

Derfor udskrives 2 3 4 (efterfulgt af et linjeskift).

De tre øverste stakelementer afstakkes, og i -værdien for det underliggende up -kald øges med 1.

up	$b = 1, c = 3, i = 3$
----	-----------------------

Nu udføres kaldet $\text{down}(4, 2)$, efterfulgt af $\text{up}(5, 1)$. Sidstnævnte kald ikke har nogen effekt, og da for -løkken for down -kaldet terminerer, øges i -værdien for det underliggende up -kald atter med 1.

up	$b = 1, c = 3, i = 4$
----	-----------------------

Nu kaldes $\text{down}(5, 2)$. Men heller ikke dette kald har nogen virkning. Algoritmen terminerer herefter.