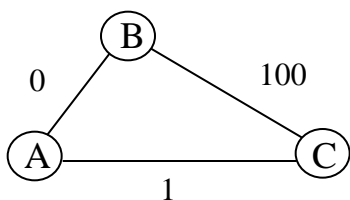


Opgaveløsninger (sæt 10)

Opgave 1a (31.5)

Betragt følgende vægtede graf:

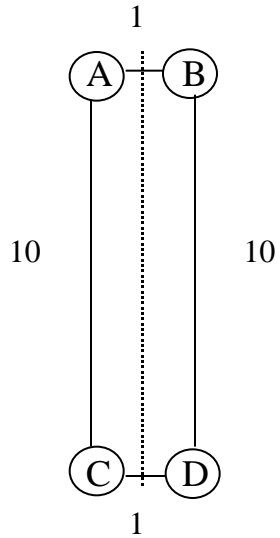


Hvis den “grådige” strategi benyttes til at bestemme den korteste vej fra A til C, bliver vejlængden $0 + 100 = 100$. Men det er let at se, at den korteste vej mellem A og C er 1.

Hvis strategien benyttes til at bestemme det minimale udspændende træ for grafen, fås værdien $0 + 100 = 100$. Men det er let at se, at værdien af et minimalt udspændende træ er $0 + 1 = 1$.

Opgave 1b (31.10)

For grafen



resulterer den i opgaven beskrevne “grådige” strategi i et udspændende træ med værdi $10 + 10 + 1 = 21$. Men det er let at indse træet bestående af kanterne AC, AB og CD udgør et minimalt udspændende træ - med værdi $10 + 1 + 1 = 12$.

Opgave 2a (32.1)

	A	B	C	D	E	F	G	H	I	J	K	L	M
A	0	1	1	1	1	1	1	1	1	0	0	0	0
B	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	1	0	0	0	0	0	0	0	0	0
F	0	0	0	1	1	0	0	0	0	0	0	0	0
G	0	0	1	1	1	0	0	1	1	0	0	0	0
H	0	0	0	0	0	0	0	0	1	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0
J	0	0	1	1	1	0	1	1	1	0	1	1	1
K	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	1	1	1	0	1	1	1	0	0	0	1
M	0	0	0	0	0	0	0	0	0	0	0	0	0

Resultatet blev kontrolleret ved kørsel af nedenstående Java-program, der benytter sig af Warshalls algoritme til bestemmelse af grafens transitive afslutning.

```
import IO.*;

class Program {
    public static void main(String args[]) {
        int a[][] = {{0,0,0,0,0,0,0,0,0,0,0,0,0,0},
                    {0,0,1,1,0,0,1,1,0,0,0,0,0,0},
                    {0,0,0,0,0,0,0,0,0,0,0,0,0,0},
                    {0,0,0,0,0,0,0,0,0,0,0,0,0,0},
                    {0,0,0,0,0,0,0,0,0,0,0,0,0,0},
                    {0,0,0,0,1,0,0,0,0,0,0,0,0,0},
                    {0,0,0,0,1,1,0,0,0,0,0,0,0,0},
                    {0,0,0,1,0,1,0,0,1,0,0,0,0,0},
                    {0,0,0,0,0,0,0,0,0,0,1,0,0,0},
                    {0,0,0,0,0,0,0,0,0,0,0,0,0,0},
                    {0,0,0,0,0,0,0,0,1,0,0,0,1,1},
                    {0,0,0,0,0,0,0,0,0,0,0,0,0,0},
                    {0,0,0,0,0,0,0,0,1,0,0,0,0,1},
                    {0,0,0,0,0,0,0,0,0,0,0,0,0,0}};

        int x, y, j, V = a.length-1;

        for (y = 1; y <= V; y++)
            for (x = 1; x <= V; x++)
                if (a[x][y] == 1)
                    for (j = 1; j <= V; j++)
                        if (a[y][j] == 1) a[x][j] = 1;
        for (x = 1; x <= V; x++) {
            for (y = 1; y <= V; y++)
                IO.print(a[x][y]);
            IO.println();
        }
    }
}
```

Opgave 2b (32.5)

M I H D E C G L K J F B A

Resultatet blev kontrolleret ved kørsel af nedenstående Java-program.

```

import IO.*;

public class Program {
    static int a[][] = {
        {0,0,0,0,0,0,0,0,0,0,0,0,0,0},
        {0,0,1,1,0,0,1,1,0,0,0,0,0,0},
        {0,0,0,0,0,0,0,0,0,0,0,0,0,0},
        {0,0,0,0,0,0,0,0,0,0,0,0,0,0},
        {0,0,0,0,0,0,0,0,0,0,0,0,0,0},
        {0,0,0,0,1,0,0,0,0,0,0,0,0,0},
        {0,0,0,0,1,1,0,0,0,0,0,0,0,0},
        {0,0,0,1,0,1,0,0,1,0,0,0,0,0},
        {0,0,0,0,0,0,0,0,0,1,0,0,0,0},
        {0,0,0,0,0,0,0,0,0,0,0,0,0,0},
        {0,0,0,0,0,0,0,0,1,0,0,0,1,1},
        {0,0,0,0,0,0,0,0,0,0,0,0,0,0},
        {0,0,0,0,0,0,0,1,0,0,0,0,0,1},
        {0,0,0,0,0,0,0,0,0,0,0,0,0,0}};

    static int id = 0, V = a.length - 1;
    static int val[] = new int[14]; }

    static void dfs() {
        for (int k = V; k >= 1; k--)
            if (val[k] == 0) visit(k);
    }

    static void visit(int k) {
        val[k] = ++id;
        for (int j = V; j >= 1; j--)
            if (a[k][j] == 1 && val[j] == 0)
                visit(j);
        IO.print(name(k) + " ");
    }

    static char name(int k)
        { return (char) ('A' + k - 1); }

    public static void main(String[] args) {
        dfs();
        IO.println();
    }
}

```