

Københavns Universitet  
Introduktion til diskret matematik og algoritmer -  
Problem set 3

Victor Vangkilde Jørgensen - kft410  
kft410@alumni.ku.dk

March 18, 2025

# Contents

<b>1</b>	<b>Question 1</b>	<b>3</b>
1.a	. . . . .	3
1.b	. . . . .	5
<b>2</b>	<b>Question 2</b>	<b>5</b>
2.a	. . . . .	5
2.b	. . . . .	9
2.c	. . . . .	9
2.d	. . . . .	9
<b>3</b>	<b>Question 3</b>	<b>10</b>
3.a	. . . . .	10
3.b	. . . . .	12
3.c	. . . . .	12
3.d	. . . . .	12

---

1

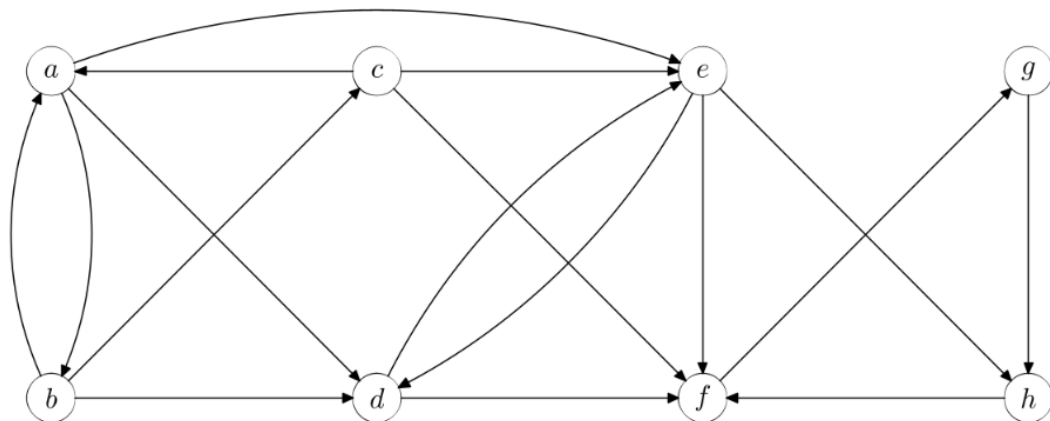


Figure 1: Directed graph  $G$  for which to compute strongly connected components in Problem 1a.

**1.a**

Vi opskriver vores directed graph som en adjacency list representation i lexicographic order:

$$a \rightarrow (b, d, e)$$

$$b \rightarrow (a, c, d)$$

$$c \rightarrow (a, e, f)$$

$$d \rightarrow (e, f)$$

$$e \rightarrow (d, f, h)$$

$$f \rightarrow (g)$$

$$g \rightarrow (h)$$

$$h \rightarrow (f)$$

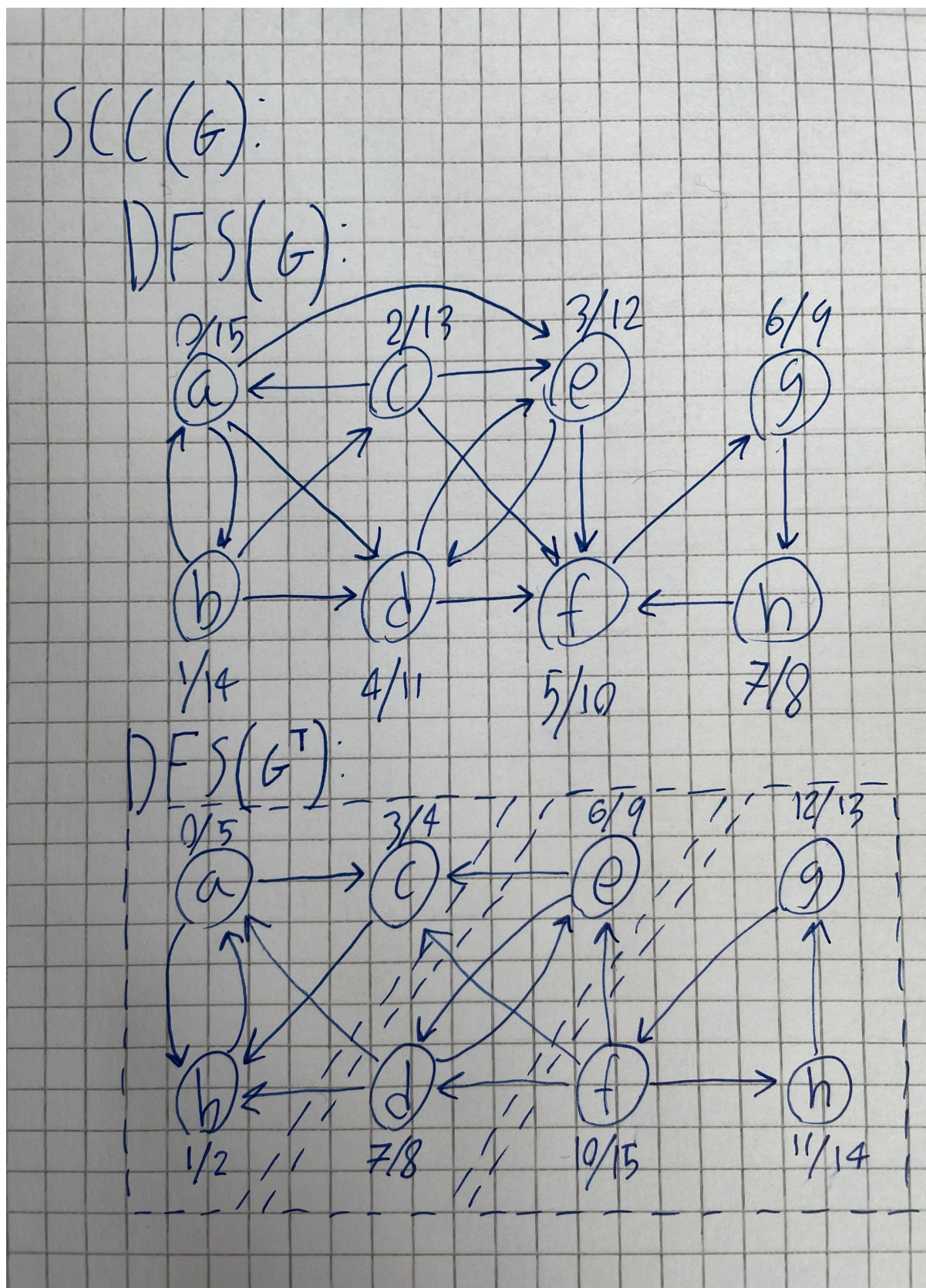


Figure 2: Gennemgang af  $SCC(G)$ .

---

1.b

2

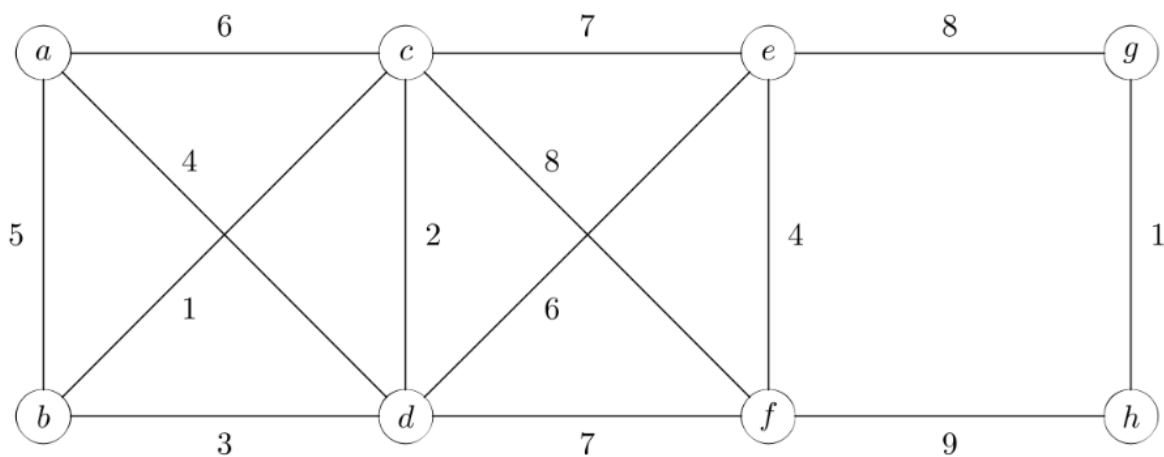


Figure 3: Undirected graph for which to compute minimum spanning tree in Problem 2a.

2.a

Vi har følgende vertices:

$a$

$c$

$e$

$g$

$b$

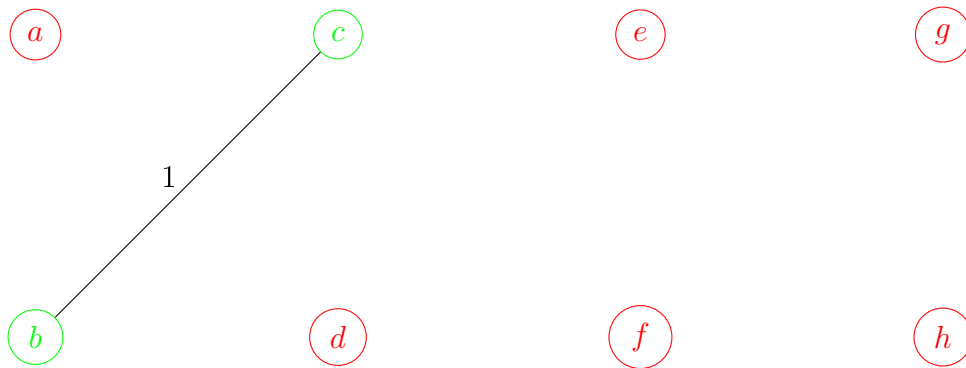
$d$

$f$

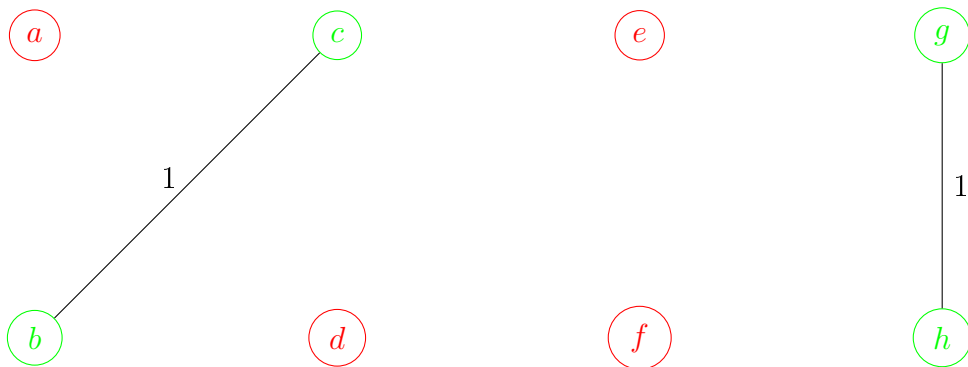
$h$

Vi ser, at  $(b, c) \in E$  har lavest vægt, sammen med  $(g, h) \in E$ . I lexicographic order vælger vi først  $(b, c) \in E$ .

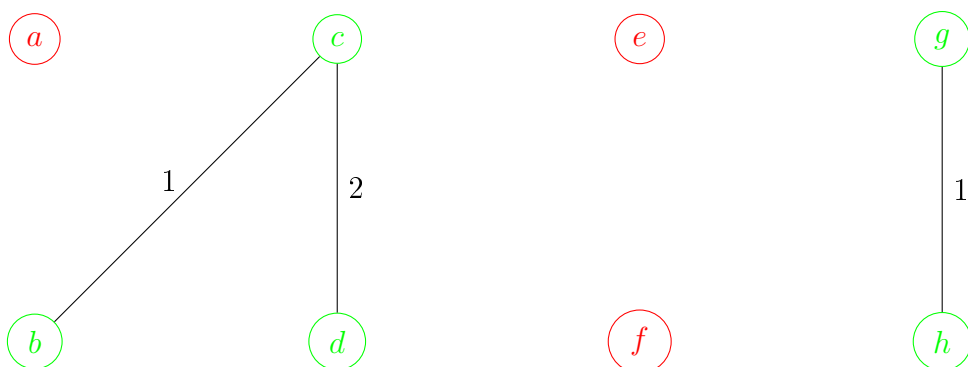
$(b, c) \in E$  connecter kun ikke-connetede vertices sammen, så vi connecter dem.



Vi kigger derefter på  $(g, h) \in E$ , da denne edge er den næste i ascending order af weight.  $(g, h) \in E$  connecter kun ikke-connectede vertices sammen, så vi connecter dem.

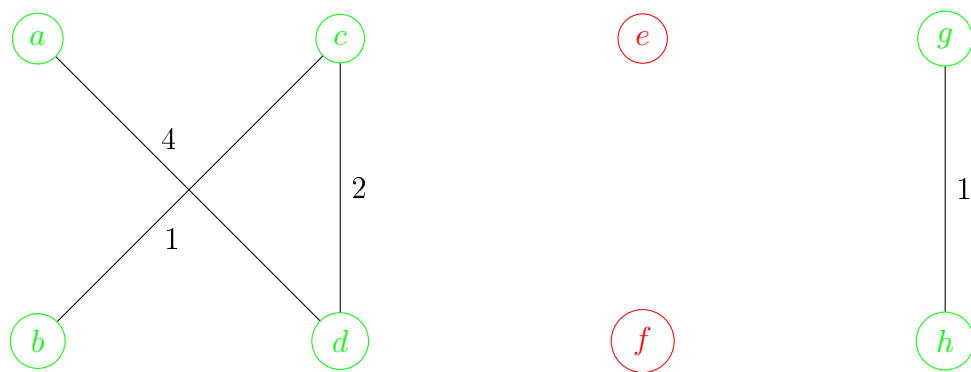


Vi kigger derefter på  $(c, d) \in E$ , da denne edge er den næste i ascending order af weight.  $(c, d) \in E$  connecter  $d$ , som ikke er connected gennem andre edges, så vi connecter dem.

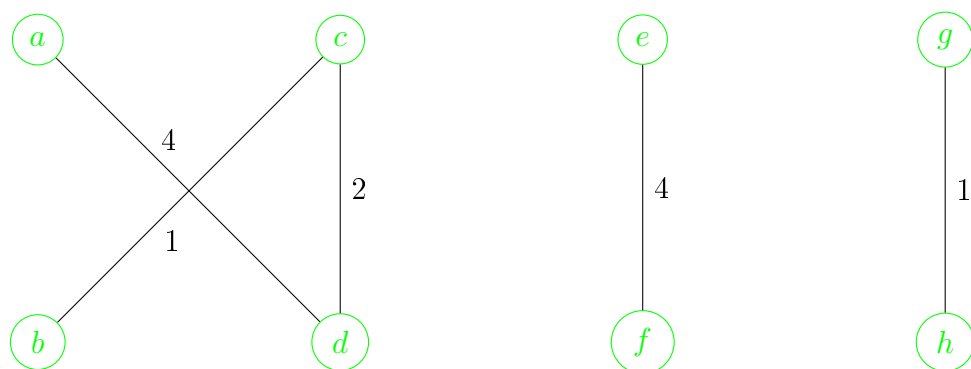


Vi kigger derefter på  $(b, d) \in E$ , da denne edge er den næste i ascending order af weight.  $b$  og  $d$ , er allerede connected gennem andre edges, så vi ignorerer den edge.

Vi kigger derefter på  $(a, d) \in E$ , da denne edge er den næste i ascending order af weight.  $(a, d) \in E$  connecter  $a$ , som ikke er connected gennem andre edges, så vi connecter dem.



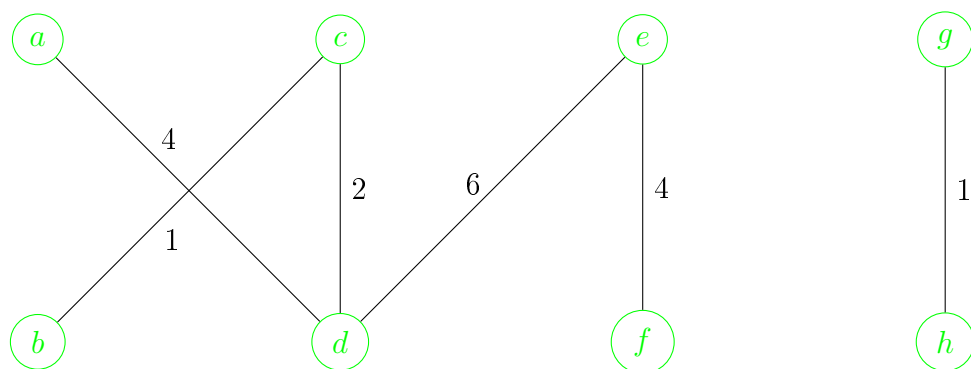
Vi kigger derefter på  $(e, f) \in E$ , da denne edge er den næste i ascending order af weight.  $(e, f) \in E$  connecter kun ikke-connectede vertices sammen, så vi connecter dem.



Vi kigger derefter på  $(a, b) \in E$ , da denne edge er den næste i ascending order af weight.  $a$  og  $b$ , er allerede connected gennem andre edges, så vi ignorerer den edge.

Vi kigger derefter på  $(a, c) \in E$ , da denne edge er den næste i ascending order af weight.  $a$  og  $c$ , er allerede connected gennem andre edges, så vi ignorerer den edge.

Vi kigger derefter på  $(d, e) \in E$ , da denne edge er den næste i ascending order af weight.  $(d, e) \in E$  connecter både  $e$  og  $f$ , som ikke er connected gennem andre edges så vi connecter dem.



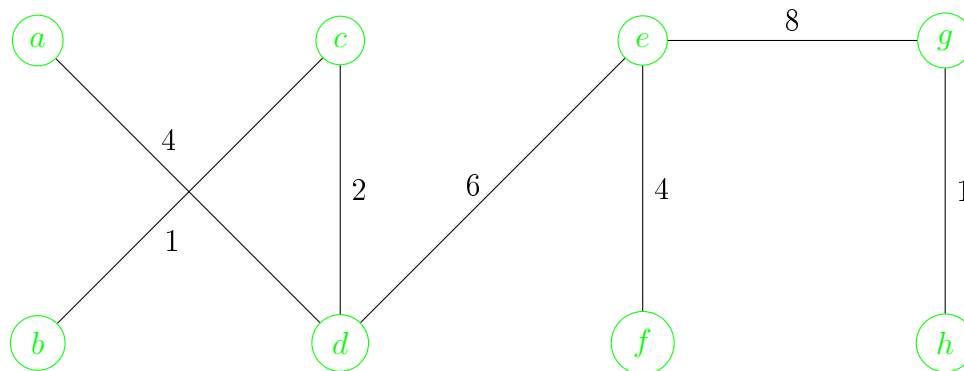
Vi kigger derefter på  $(c, e) \in E$ , da denne edge er den næste i ascending order af weight.  $c$  og  $e$ , er allerede connected gennem andre edges, så vi ignorerer den edge.

---

Vi kigger derefter på  $(d, f) \in E$ , da denne edge er den næste i ascending order af weight.  $d$  og  $f$ , er allerede connected gennem andre edges, så vi ignorerer den edge.

Vi kigger derefter på  $(c, f) \in E$ , da denne edge er den næste i ascending order af weight.  $c$  og  $f$ , er allerede connected gennem andre edges, så vi ignorerer den edge.

Vi kigger derefter på  $(e, g) \in E$ , da denne edge er den næste i ascending order af weight.  $(e, g) \in E$  connecter både  $e$  og  $g$ , som ikke er connected gennem andre edges så vi connecter dem.



Vi ser, at antallet af edges svarer til antallet af vertices - 1. Vi er dermed færdige, da vi har 7 edges og 8 vertices.



---

2.b

2.c

2.d

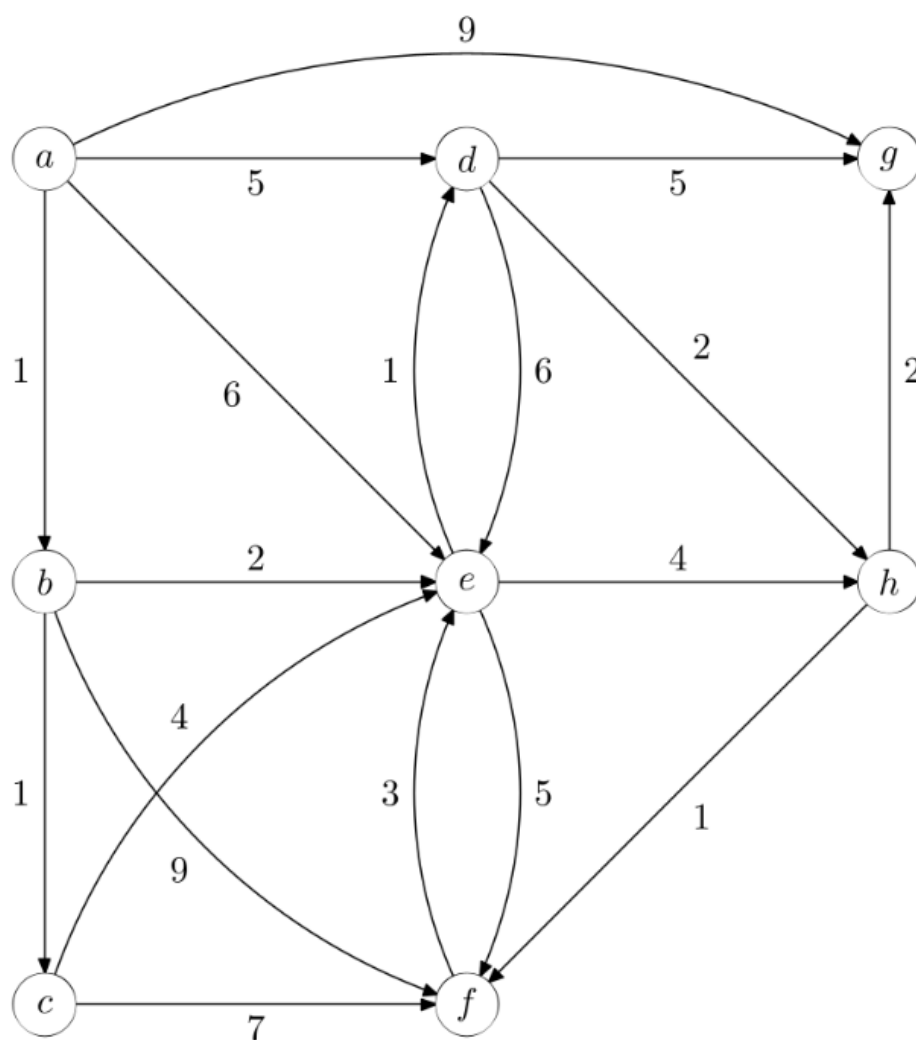


Figure 4: Directed graph Dijkstra's algorithm in Problem 3a.

---

### 3

#### 3.a

Jeg opskriver vores weighted directed graph som adjacency list representation (*vertex, weight*):

$$\begin{aligned}(a, 0) &\rightarrow ((b, 1), (d, 5), (e, 6), (g, 9)) \\(b, 0) &\rightarrow ((c, 1), (e, 2), (f, 9)) \\(c, 0) &\rightarrow ((e, 4), (f, 7)) \\(d, 0) &\rightarrow ((e, 6), (g, 5), (h, 2)) \\(e, 0) &\rightarrow ((f, 5), (h, 4)) \\(f, 0) &\rightarrow (e, 3) \\(g, 0) &\rightarrow () \\(h, 0) &\rightarrow (f, 1)\end{aligned}$$

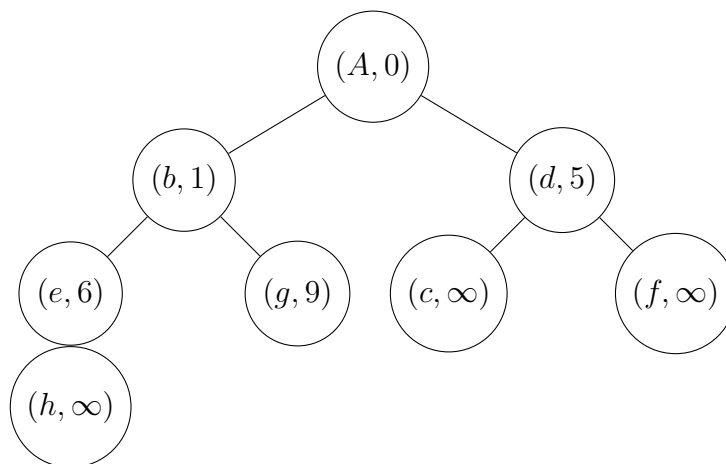
Udregning af distancer for  $a$ 's naboer:

$$[(b, 1), (d, 5), (e, 6), (g, 9)]$$

Vi opdaterer nu de laveste værdier i grafen:

$$[(A, 0), (b, 1), (c, \infty), (d, 5), (e, 6), (f, \infty), (g, 9), (h, \infty)]$$

$a$  er nu besøgt, så det har jeg markeret ved at skrive  $A$  i stedet.



Vi ser, at  $b$  er den ubesøgte vertex med kortest afstand fra  $a$ .

Udregning af nye distancer for  $b$ 's naboer:

$$[(c, 1 + 1), (e, 1 + 2), (f, 1 + 9)] = [(c, 2), (e, 3), (f, 10)]$$

Vi opdaterer nu de laveste værdier i grafen:

$$[(A, 0), (B, 1), (c, 2), (e, 3), (d, 5), (g, 9), (f, 10), (h, \infty)]$$

$b$  er nu besøgt, så det har jeg markeret ved at skrive  $B$  i stedet.

---

Vi ser, at  $c$  er den ubesøgte vertex med kortest afstand fra  $a$ .

Udregning af nye distancer for  $c$ 's naboer:

$$[(e, 2 + 4), (f, 2 + 7)] = [(e, 6), (f, 9)]$$

Vi opdaterer nu de laveste værdier i grafen:

$$[(A, 0), (B, 1), (C, 2), (e, 3), (d, 5), (f, 9), (g, 9), (h, \infty)]$$

$c$  er nu besøgt, så det har jeg markeret ved at skrive  $C$  i stedet.

Vi ser, at  $e$  er den ubesøgte vertex med kortest afstand fra  $a$ .

Udregning af nye distancer for  $e$ 's naboer:

$$[(f, 3 + 5), (h, 3 + 4)] = [(f, 8), (h, 7)]$$

Vi opdaterer nu de laveste værdier i grafen:

$$[(A, 0), (B, 1), (C, 2), (E, 3), (d, 5), (h, 7), (f, 8), (g, 9)]$$

$e$  er nu besøgt, så det har jeg markeret ved at skrive  $E$  i stedet.

Vi ser, at  $d$  er den ubesøgte vertex med kortest afstand fra  $a$ .

Udregning af nye distancer for  $d$ 's naboer:

$$[(g, 5 + 5), (h, 5 + 5), (e, 5 + 6)] = [(g, 10), (h, 10), (e, 11)]$$

Vi opdaterer nu de laveste værdier i grafen, men ser, at der ikke nogen kortere afstande:

$$[(A, 0), (B, 1), (C, 2), (E, 3), (D, 5), (h, 7), (f, 8), (g, 9)]$$

$d$  er nu besøgt, så det har jeg markeret ved at skrive  $D$  i stedet.

Vi ser, at  $h$  er den ubesøgte vertex med kortest afstand fra  $a$ .

Udregning af nye distancer for  $h$ 's naboer:

$$[(f, 7 + 1)] = [(f, 8)]$$

Vi opdaterer nu de laveste værdier i grafen, men ser, at der ikke nogen kortere afstande:

$$[(A, 0), (B, 1), (C, 2), (E, 3), (D, 5), (H, 7), (f, 8), (g, 9)]$$

$h$  er nu besøgt, så det har jeg markeret ved at skrive  $H$  i stedet.

Vi ser, at  $f$  er den ubesøgte vertex med kortest afstand fra  $a$ .

Udregning af nye distancer for  $f$ 's naboer:

$$[(e, 8 + 3)] = [(e, 11)]$$

Vi opdaterer nu de laveste værdier i grafen, men ser, at der ikke nogen kortere afstande:

$$[(A, 0), (B, 1), (C, 2), (E, 3), (D, 5), (H, 7), (F, 8), (g, 9)]$$

$f$  er nu besøgt, så det har jeg markeret ved at skrive  $F$  i stedet.

Vi ser, at  $g$  er den sidste ubesøgte vertex, og at  $g$  ikke har nogen naboer.

$$[(A, 0), (B, 1), (C, 2), (E, 3), (D, 5), (H, 7), (F, 8), (G, 9)]$$

Vi markerer  $g$  som besøgt med  $G$ .

---

**3.b**

**3.c**

**3.d**