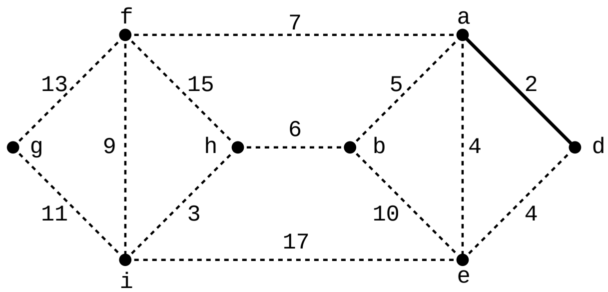
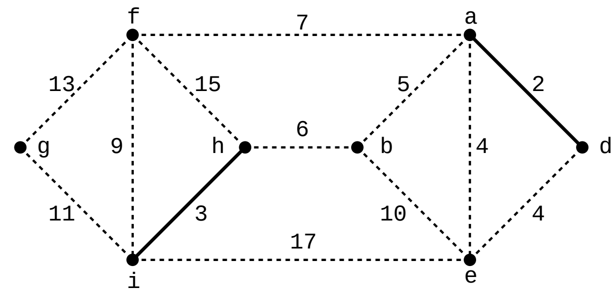
学籍番号		氏名	

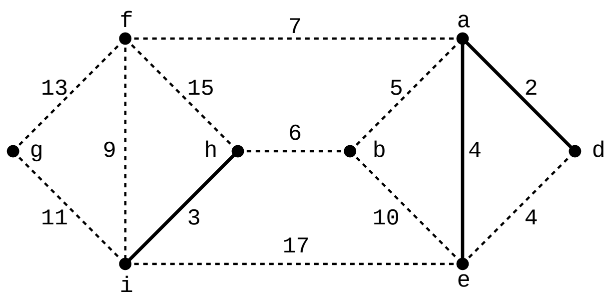
問題 1(1) の解答欄



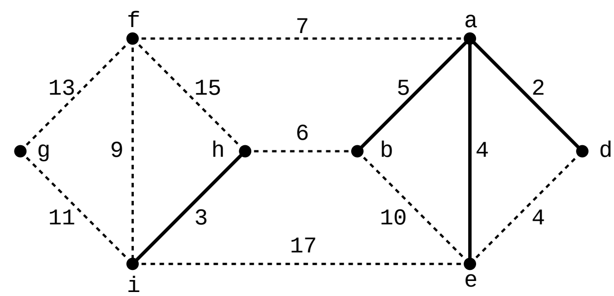
1 本目の枝が入った時点



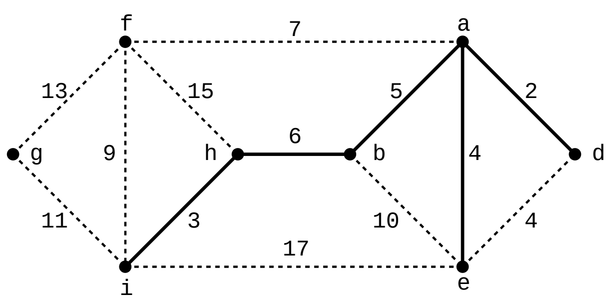
2 本目の枝が入った時点



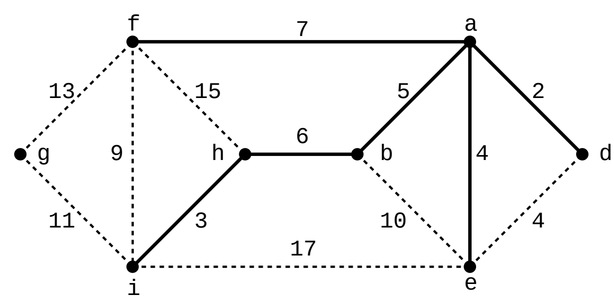
3 本目の枝が入った時点



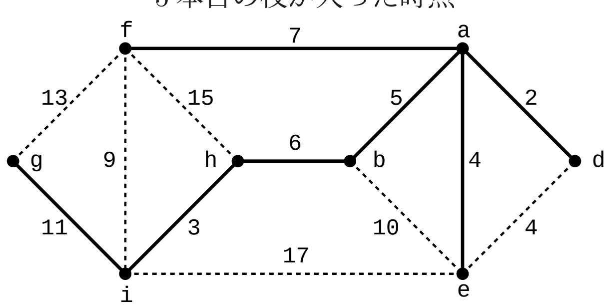
4 本目の枝が入った時点



5 本目の枝が入った時点



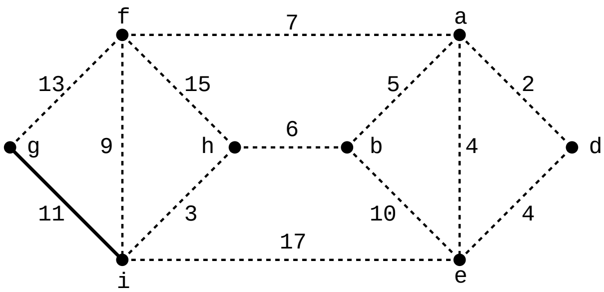
6 本目の枝が入った時点



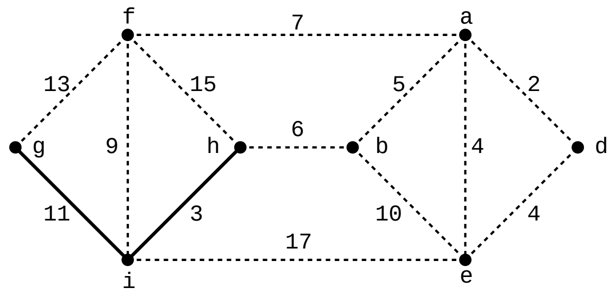
7 本目の枝が入った時点

学籍番号		氏名	

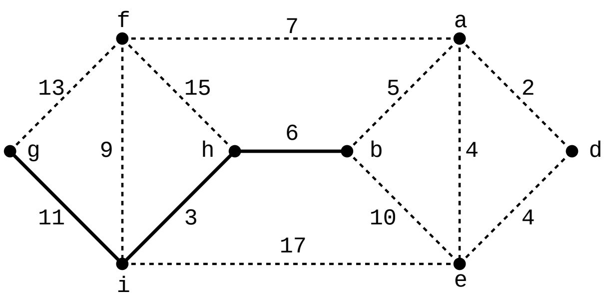
問題 1(2) の解答欄



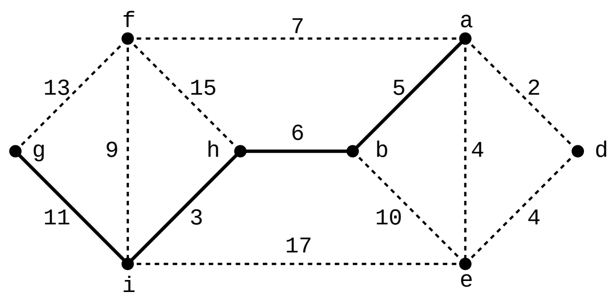
1 本目の枝が入った時点



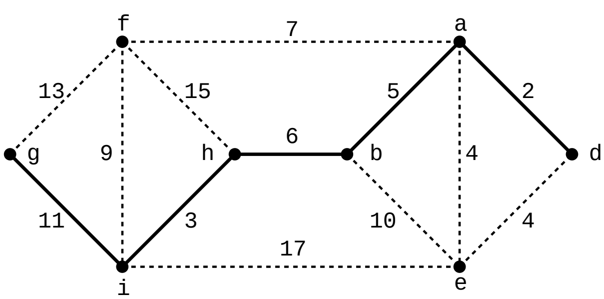
2 本目の枝が入った時点



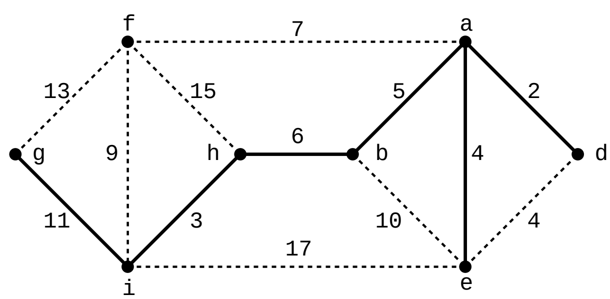
3 本目の枝が入った時点



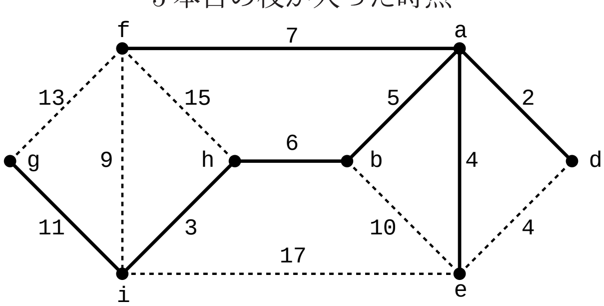
4 本目の枝が入った時点



5 本目の枝が入った時点



6 本目の枝が入った時点



7 本目の枝が入った時点

学籍番号		氏名	

問題 2(1) の解答欄

ある $A \subseteq V$ に対して, $\delta(A)$ と表される枝部分集合. ここで, $\delta(A)$ は片方の終点を A の中に持ち, もう片方の終点を A の外に持つような枝全体である.

問題 2(2) の解答欄

$G = (V, E)$ の連結で閉路を含まない部分グラフで, ノード集合が V であるもの.

問題 2(3) の解答欄

G の全域木の中で, その重みが最小であるもの. ここで, 全域木 $H = (V, F)$ の重みは, $\sum_{e \in F} c_e$ で定義される.

問題 2(4) の解答欄

$A \subseteq B$ となるような最小全域木 $H = (V, B)$ が存在すること.

問題 2(5) の解答欄

$H = (V, T)$ を $B \subseteq T$ であるような最小全域木とする. B の定義からそのような全域木は存在する. もし $e \in T$ であるのならば,

$$B \cup \{e\} \subseteq T$$

であるから, $B \cup \{e\}$ も最小全域木に拡張可能である.

$e \notin T$ のときを考えよう. $e = vw$ として, P を H の中の v から w への道とする. D はカットであるから, $f \in D$ となる P の枝 f がある. $e \in D$ の定義によって, $c_e \leq c_f$ である. また補題 2.7 より, $(V, (T \cup \{e\}) \setminus \{f\})$ も木である. さらに,

$$c((T \cup \{e\}) \setminus \{f\}) = c(T) + c_e - c_f \leq c(T)$$

であるから, $(V, (T \cup \{e\}) \setminus \{f\})$ は最小全域木である.

$B \cup \{e\} \subseteq (T \cup \{e\}) \setminus \{f\}$ なので, $B \cup \{e\}$ は最小全域木に拡張可能である.

学籍番号		氏名	

問題 3(1) の解答欄

(a) このグラフは非巡回的であるから, 位相的順序 $a \to g \to f \to h \to d \to i \to j \to e \to b$ の順に各点を一回だけスキャンすれば良い.

(b)

表 1: アルゴリズムの進行状況

	Start		$vw = ag$		$vw = ai$		$vw = gf$		$vw = gd$		$vw = fh$	
	y	p	y	p	y	p	y	p	y	p	y	p
a	0	0	0	0	0	0	0	0	0	0	0	0
b	∞	-1	∞	-1	∞	-1	∞	-1	∞	-1	∞	-1
d	∞	-1	∞	-1	∞	-1	∞	-1	-4	g	-4	g
e	∞	-1	∞	-1	∞	-1	∞	-1	∞	-1	∞	-1
f	∞	-1	∞	-1	∞	-1	1	g	1	g	1	g
g	∞	-1	-3	a	-3	a	-3	a	-3	a	-3	a
h	∞	-1	∞	-1	∞	-1	∞	-1	∞	-1	4	f
i	∞	-1	∞	-1	-2	a	-2	a	-2	a	-2	a
j	∞	-1	∞	-1	∞	-1	∞	-1	∞	-1	∞	-1

表 2: アルゴリズムの進行状況 (続き)

	$vw = hb$		$vw = de$		$vw = ij$		$vw = eb$		$vw =$		$vw =$	
	y	p	y	p	y	p	y	p	y	p	y	p
a	0	0	0	0	0	0	0	0				
b	1	h	1	h	1	h	-4	e				
d	-4	g	-4	g	-4	g	-4	g				
e	∞	-1	-6	d	-6	d	-6	d				
f	1	g	1	g	1	g	1	g				
g	-3	a	-3	a	-3	a	-3	a				
h	4	f	4	f	4	f	4	f				
i	-2	a	-2	a	-2	a	-2	a				
j	∞	-1	∞	-1	-1	i	-1	i				

学籍番号		氏名	

表 3: アルゴリズムの進行状況 (続き)

	$vw =$		$vw =$		$vw =$		$vw =$		$vw =$		$vw =$	
	y	p	y	p	y	p	y	p	y	p	y	p
a												
b												
d												
e												
f												
g												
h												
i												
j												

問題 3(2) の解答欄

(a) 閉路はあるが, 全ての枝のコストは非負だからダイクストラ法を用いる.

(b)

表 4: アルゴリズムの進行状況

	Start		$vw = ag$		$vw = ai$		$vw = gf$		$vw = gd$		$vw = dh$	
	y	p	y	p	y	p	y	p	y	p	y	p
a	0	0	0	0	0	0	0	0	0	0	0	0
b	∞	-1	∞	-1	∞	-1	∞	-1	∞	-1	∞	-1
d	∞	-1	∞	-1	∞	-1	∞	-1	3	g	3	g
e	∞	-1	∞	-1	∞	-1	∞	-1	∞	-1	∞	-1
f	∞	-1	∞	-1	∞	-1	5	g	5	g	5	g
g	∞	-1	1	a	1	a	1	a	1	a	1	a
h	∞	-1	∞	-1	∞	-1	∞	-1	∞	-1	7	d
i	∞	-1	∞	-1	5	a	5	a	5	a	5	a
j	∞	-1	∞	-1	∞	-1	∞	-1	∞	-1	∞	-1

学籍番号		氏名	

表 5: アルゴリズムの進行状況 (続き)

	$vw = de$		$vw = di$		$vw = ij$		$vw = fh$		$vw = je$		$vw = hb$	
	y	p	y	p	y	p	y	p	y	p	y	p
a	0	0	0	0	0	0	0	0	0	0	0	0
b	∞	-1	∞	-1	∞	-1	∞	-1	∞	-1	9	h
d	3	g	3	g	3	g	3	g	3	g	3	g
e	10	d	10	d	10	d	10	d	6	j	6	j
f	5	g	5	g	5	g	5	g	5	g	5	g
g	1	a	1	a	1	a	1	a	1	a	1	a
h	7	d	7	d	7	d	6	f	6	f	6	f
i	5	a	4	d	4	d	4	d	4	d	4	d
j	∞	-1	∞	-1	5	i	5	i	5	i	5	i

表 6: アルゴリズムの進行状況 (続き)

	$vw = eb$		$vw =$		$vw =$		$vw =$		$vw =$		$vw =$	
	y	p	y	p	y	p	y	p	y	p	y	p
a	0	0										
b	6	e										
d	3	g										
e	6	j										
f	5	g										
g	1	a										
h	6	f										
i	4	d										
j	5	i										

学籍番号		氏名	

問題 3(3) の解答欄

(a) 閉路が存在し, 負のコストの枝もあるため, フォード-ベルマンのアルゴリズムを用いる. ノードをスキャンすることによって枝を調べる. ポテンシャルが更新された頂点をキューに保存し, 次にスキャンするノードをキューから取り出す. キューには最初は a だけが入っている.

(b)

表 7: アルゴリズムの進行状況

	Start		$vw=ag$		$vw=ai$		$vw=gd$		$vw=ij$		$vw=id$	
	y	p	y	p	y	p	y	p	y	p	y	p
a	0	0	0	0	0	0	0	0	0	0	0	0
b	∞	-1	∞	-1	∞	-1	∞	-1	∞	-1	∞	-1
d	∞	-1	∞	-1	∞	-1	0	g	0	g	-2	i
e	∞	-1	∞	-1	∞	-1	∞	-1	∞	-1	∞	-1
f	∞	-1	∞	-1	∞	-1	∞	-1	∞	-1	∞	-1
g	∞	-1	1	a	1	a	1	a	1	a	1	a
h	∞	-1	∞	-1	∞	-1	∞	-1	∞	-1	∞	-1
i	∞	-1	∞	-1	-3	a	-3	a	-3	a	-3	a
j	∞	-1	∞	-1	∞	-1	∞	-1	-1	i	-1	i

表 8: アルゴリズムの進行状況 (続き)

	$vw=dh$		$vw=de$		$vw=hf$		$vw=hb$		$vw=fg$		$vw=be$	
	y	p	y	p	y	p	y	p	y	p	y	p
a	0	0	0	0	0	0	0	0	0	0	0	0
b	∞	-1	∞	-1	∞	-1	-2	h	-2	h	-2	h
d	-2	i	-2	i	-2	i	-2	i	-2	i	-2	i
e	∞	-1	0	d	0	d	0	d	0	d	-4	b
f	∞	-1	∞	-1	-3	h	-3	h	-3	h	-3	h
g	1	a	1	a	1	a	1	a	-1	f	-1	f
h	-1	d	-1	d	-1	d	-1	d	-1	d	-1	d
i	-3	a	-3	a	-3	a	-3	a	-3	a	-3	a
j	-1	i	-1	i	-1	i	-1	i	-1	i	-1	i

学籍番号		氏名	

表 9: アルゴリズムの進行状況 (続き)

	$vw = ej$		$vw =$		$vw =$		$vw =$		$vw =$		$vw =$	
	y	p	y	p	y	p	y	p	y	p	y	p
a	0	0										
b	-2	h										
d	-2	i										
e	-4	b										
f	-3	h										
g	-1	f										
h	-1	d										
i	-3	a										
j	-5	e										

学籍番号		氏名	

問題 4(1) の解答欄

$$y_v + c_{vw} \geq y_w$$

が全ての枝 $vw \in E$ について成り立ち, かつ, $y_r = 0$ であること.

問題 4(2) の解答欄

y を (G, c) の実行可能ポテンシャルとする.

$$v_0, e_1, v_1, e_2, \cdots, v_{k-1}, e_k, v_k (= v_0)$$

を G の任意な閉路とすると, そのコストは

$$\sum_{i=1}^{k} c_{e_i} \geq \sum_{i=1}^{k} (y_{v_i} - y_{v_{i-1}}) = y_{v_k} - y_{v_0} = 0.$$

ゆえに, 負のコストを持つ閉路は存在しない.

問題 4(3) の解答欄

(G, c) には負のコストの閉路は存在しないと仮定する.

G に新しいノード s を付け加えて, s から G の全てのノードへコスト 0 の枝を張る. こうしてできたネットワークを (G', c') とする.

(G, c) には負のコストの閉路は存在しないから, (G', c') にも負のコストの閉路は存在しない. なぜならば, s を通る G' の閉路は存在しないからである.

Ford のアルゴリズムを (G', c') と始点 s に対して適用すれば, 定理 2.11 によって (G', c') の実行可能ポテンシャル y'_v $(v \in V')$ が得られる.

y' を V に制限したベクトルを y とすると, $y_v + y_{vw} \geq y_w$ $(vw \in E)$ が成り立つ. もし必要ならば, $y_v \leftarrow y_v - y_r$ $(v \in V)$ とすることによって, y は (G, c) の実行可能ポテンシャルとなる.