学籍番号		氏名	

問題 1(1) の解答欄 (配点 8)

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

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 1(2) の解答欄 (配点 8)

r から v への任意の有向道

$$P = v_0, e_1, v_1, \cdots, e_k, v_k$$

を考える. ここで, $v_0 = r, v_k = v$ である. y は, feasible potential であるから, $y_{v_{i-1}} + c_{e_i} \geq y_{v_i}$ がすべての $i = 1, \cdots, k$ について成り立つ. したがって,

$$\begin{aligned} c(P) &= \sum_{i=1}^{k} c_{e_i} \\ &\geq \sum_{i=1}^{k} y_{v_i} - y_{v_{i-1}} \\ &= y_{v_k} - y_{v_0} \\ &= y_v - y_r \\ &= y_v. \end{aligned}$$

問題 1(3) の解答欄 (配点 10)

r から v への単純な有向道は有限個しかない. したがって, Proposition 2.10(i) によって, 各 $v \in V$ に対して y_v が取り得る値の数も有限個しかない. その一方で, Ford のアルゴリズムの各繰り返しにおいて、少くとも 1 つの y_v が減少する. (そして、増加する y_v は無い.) したがって, アルゴリズムはいつかは停止する.

終了時において, Proposition 2.10(ii) より p は、r から v への長さが高々y_v の有向道を定義する。Proposition 2.9 によって, この有向道は最短である.

学籍番号		氏名	

問題 1(4) の解答欄 (配点 10)

(G, c) が feasible potential y を持つと仮定する。任意の dicircuit

$$D = v_0, e_1, v_1, \cdots, e_k, v_k = v_0$$

を考えると、その費用は

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

ゆえに、負の費用を持つ dicircuit は存在しない.

逆に、(G, c) には負の費用を持つ dicircuit は存在しないと仮定する. G に新しいノード r を追加して、各 $v \in V$ に対して、費用 0 の枝 rv を追加する。このように定義された有向グラフを G' として、その費用ベクトルを c' とすると, (G', c') には負の費用を持つ dicircuit は存在しない。したがって、Ford のアルゴリズムを (G', c') に対して適用すれば、Theorem 2.11 によって、それは終了し、feasible potential y' を出力する。y' の成分をもとのグラフのノードに制限すれば、(G, c) の feasible potential y を得る.

問題 2(1) の解答欄 (配点 12)

初期化は、O(n) 時間で実行できる。

各 i 回目のパスにおいて、全ての枝について、基本ステップ「correct かどうかの判定を行い、incorrect ならば、correct する」を行う。基本ステップは O(1) 時間で実行できるので、1 回のパスには O(m) 時間かかる. i は 0 から最大 n まで増加する可能性があるのでパスの総数は最大 n である。したがって、アルゴリズム全体では O(n)+O(mn)=O(mn) 時間かかる。

学籍番号		氏名	

問題 2(2) の解答欄 (配点 20)

表 2.1: Ford-Bellman のアルゴリズムの動き.

i		a	g	i	d	h	e	b
0	y	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	0
	p	-1	-1	-1	-1	-1	-1	0
1	y	-1	0	-3	-2	1	-1	0
	p	i	d	e	i	e	b	0
2	y	-1	0	-3	-2	1	-1	0
	p	i	d	e	i	e	b	0
3	y							
	p							
4	y							
	p							
5	y							
	p							
6	y							
	p							
7	y							
	p							
8	y							
	p							

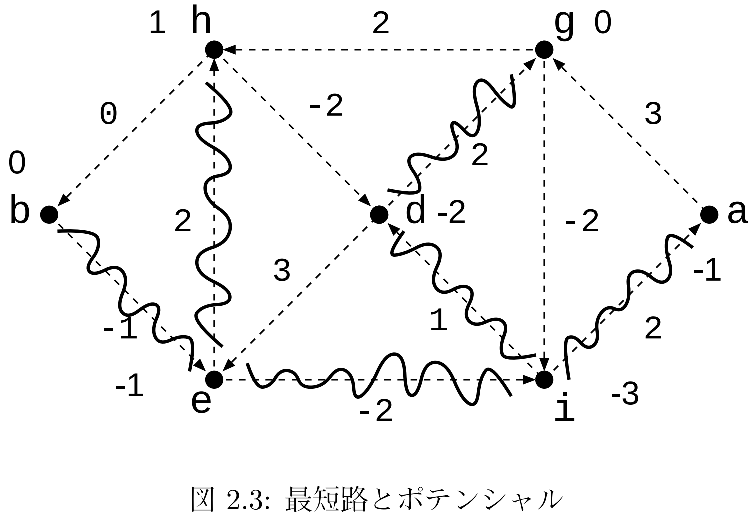

図 2.3: 最短路とポテンシャル

学籍番号		氏名	

問題 3(1) の解答欄 (配点 12)

初期化は、O(n) 時間で実行できる。

While の各繰り返して y_v が最小であるような v の発見に O(n) かかり、S から v を削除するために O(1) かかる。v の Scan には、v から出る枝の本数分だけ時間がかかる。したがって、アルゴリズム全体では、

$$\begin{aligned} & \mathrm{O}(n) + \sum_{v \in V} (\mathrm{O}(n) + \mathrm{O}(1) + \mathrm{O}(v\text{ から出る枝の本数})) \\ = \; & \mathrm{O}(n) + \mathrm{O}(n^2) + \mathrm{O}(n) + \mathrm{O}(m) \\ = \; & \mathrm{O}(n^2) \end{aligned}$$

時間かかる。

学籍番号		氏名	

問題 3(2) の解答欄 (配点 20)

表 2.2: Dijkstra のアルゴリズムの動き.

スキャンされたノード		i	f	d	j	h	k	b
	y	$+\infty$	$+\infty$	0^*	$+\infty$	$+\infty$	$+\infty$	$+\infty$
	p	-1	-1	0	-1	-1	-1	-1
d	y	$+\infty$	2^*	0	7	$+\infty$	$+\infty$	8
	p	-1	d	0	d	-1	-1	d
f	y	$+\infty$	2	0	7	3^*	$+\infty$	7
	p	-1	d	0	d	f	-1	f
h	y	10	2	0	7	3	$+\infty$	5^*
	p	h	d	0	d	f	-1	h
b	y	9	2	0	6^*	3	7	5
	p	b	d	0	b	f	b	h
j	y	9	2	0	6	3	7^*	5
	p	b	d	0	b	f	b	h
k	y	8^*	2	0	6	3	7	5
	p	k	d	0	b	f	b	h
i	y	8	2	0	6	3	7	5
	p	k	d	0	b	f	b	h

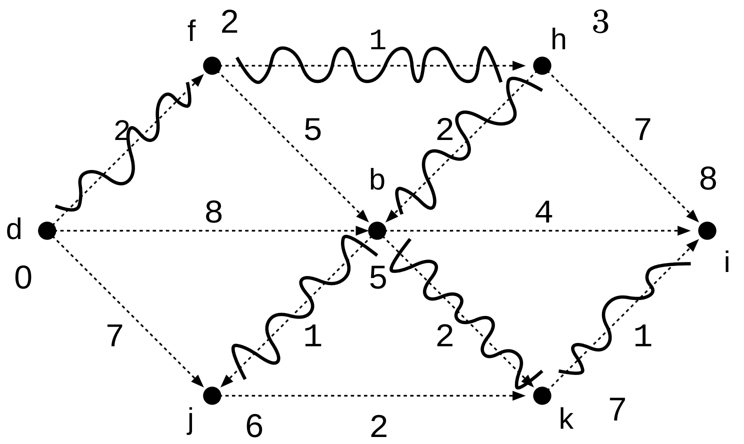

図 2.4: 最短路とポテンシャル