井上研究室　2016 年度院試回答

大問 1：アルゴリズムとプログラミング-嶋利

大問 2：計算機システムとシステムプログラム-瀬村

大問 4 計算理論-横井

大問 6:電子回路と論理設計-岡島

大問 3 :離散構造-徳井

(1)
0 2 5 255 (0 から各頂点への直接の距離)
0 2 3 3 (0 から 1 を経由してもいいという条件においての、0 から各頂点への最短距離)

(2)
O(n^2)
25 行目の比較文を頂点の数の二乗回行う必要がある．処理の最大回数はこの比較文を実行した回数の高々定数倍である．

(3)
(ア) Prev[j]=i;
26 行目の比較は例をあげると『0 から 2 への距離”5”が，0 から 1 を経由して 2 へ行く距離”3”よりも長いかどうか？』を示す．したがって，その際に j は 2，i には 1 が入っている．この際に (ア) を実行すると元々Prev[2]=0 だったものが Prev[2]=1 となり，最短経路が更新されている．同様の操作を行ってやれば各頂点への最短経路を残せる

(イ) printpath(3)
終点の頂点を与えてやれば再帰的に (ア) で登録した情報を呼び出し，始点から終点への経路が出せる

(4-1)
3 回 compute(w,n,0,3)，compute(w,n,1,3)，compute(w,n,2,3) を呼びだしてやればよい．compute(w,n,0,3) を例に挙げると，呼び出しの Len[0]〜Len[3] には 0 から各頂点への最短経路が入っている。これを利用すれば上記の三回ですべて呼び出せる．問題文に同一頂点なら最短経路長 0 とあるので，compute(w,n,3,3) は呼び出さなくてよいと考えられる）参考：

```
for (l = 0; l < 3; l++) {
    compute(w, n, l, 3);
    printf("%d %d %d %d\n",Len[0], Len[1], Len[2], Len[3]);
}
```

で

```
0 2 3 3
2 0 1 1
3 1 0 2
```

が出力される．これの右上半分を使えばよい．
難 (4-2)　(4-1) と同様にすれば三回で呼び出せる．
コードを載せているので参考にしてください．
文責 嶋利備考：出題ミスがあったため，全員に 50 点分入りました。主観ですが，ほかの年度に比べて対策しにくい分しんどかったです。

```
#include <stdio.h>
#define INF 255
#define MAXN 16
int Len[MAXN]; int Prev[MAXN]; void printpath(int v);
int allvisited(int *a, int n) {
  int i, r = 1;
  for (i = 0; i < n; i++) { r *= a[i]; }
    return r;
}
void compute(int *w, int n, int s, int d) {
  int i, j, next, min, visited[MAXN];
  for (i = 0; i < n; i++) {
    Len[i] = INF; visited[i] = 0;
  }
  i = s; Len[i] = 0; visited[i] = 1; Prev[i] = -1;
  //while (visited[d] == 0) {
  while (allvisited(visited, n) == 0) {
    min = INF; next = d;
    for (j = 0; j < n; j++) {
      if (visited[j] == 1) continue;
      if (Len[j] > Len[i] + w[i*n + j]) {
        Len[j] = Len[i] + w[i*n + j];      Prev[j] = i;
      }
      if (min > Len[j]) {
        min = Len[j]; next = j;
      }
    }
    i = next; visited[i] = 1;
  }
}

int main() {
  int w[] = { 0,2,5,INF,
              2,0,1,1,
              5,1,0,2,
              INF,1,2,0 };
  int n = 4;
  int l;
  for (l = 0; l < 4; l++) {
    compute(w, n, l, 3);
    printf("%d %d %d %d\n",Len[0], Len[1], Len[2], Len[3]);
  }
  printf("Shortest Path Length from 0 to 3 is %d\n",Len[3]);
  printpath(3); printf("\n");
}

void printpath(int v) {
if (v != -1) {
  printpath(Prev[v]); printf(" - %d ", v); }
}
```

大問 **2**：計算機システムとシステムプログラム

(1-1-1)

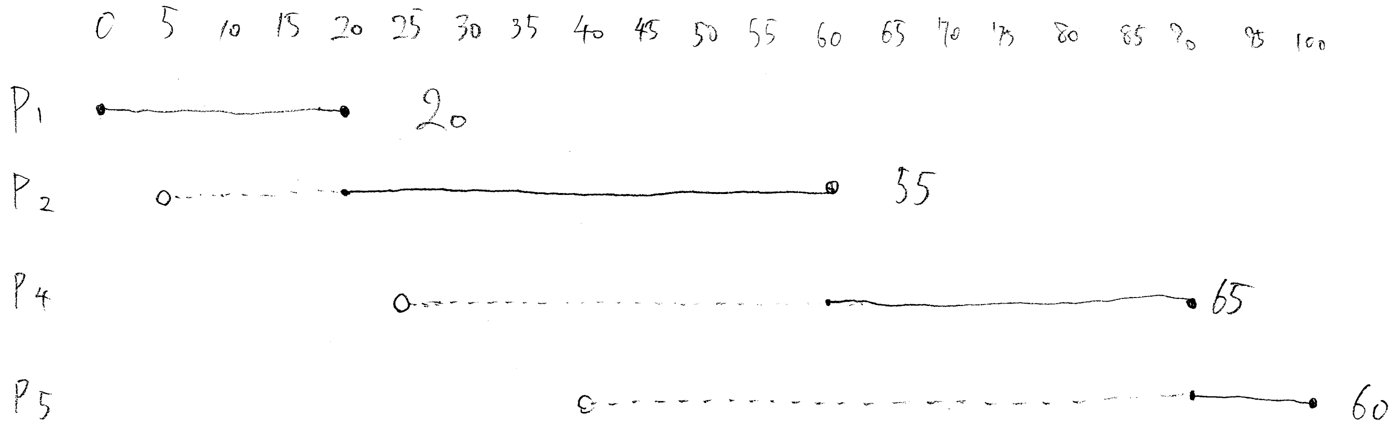
	最小値	のビット列	最大値	のビット列
符号なし整数	0	0000000000	1023	1111111111
符号絶対値表現	-511	1111111111	511	0111111111
1 の補数表現	-511	1000000000	511	0111111111
2 の補数表現	-512	1000000000	511	0111111111

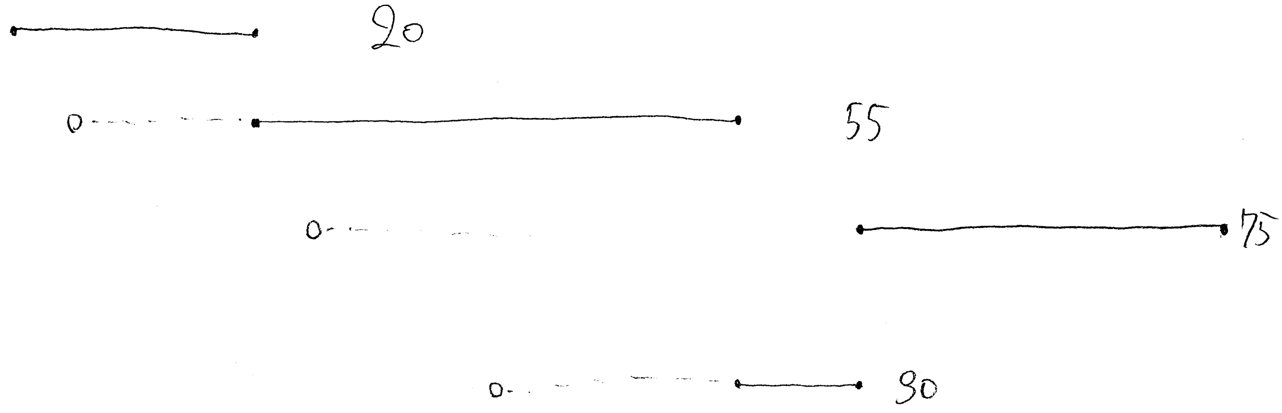
(1-1-2)
符号絶対値表現： 1000110100
1 の補数表現： 1111001011
2 の補数表現： 1111001100
(1-1-3)
符号なし整数： 691
符号絶対値表現： -179
1 の補数表現： -332
2 の補数表現： -333
(1-2-1)
$s_i = \overline{a_i}\overline{b_i}c_i + a_i\overline{b_i}\overline{c_i} + \overline{a_i}b_i\overline{c_i} + a_ib_ic_i$, $c_{i+1} = a_ib_i + b_ic_i + a_ic_i$
(1-2-2)
$c_3 = c_0p_0p_1p_2 + g_0p_1p_2 + g_1p_2 + g_2$
(1-2-3)
4+4+4+6=18(ns) 例：$a_0 - c_1 - c_2 - c_3 - s_3$
2+6+6=14(ns) 例：$a_0 - g_0 - c_3 - s_3$
(2-1)
(ア) a (イ) k (ウ) c (エ) g (オ) h (カ) d (キ) l

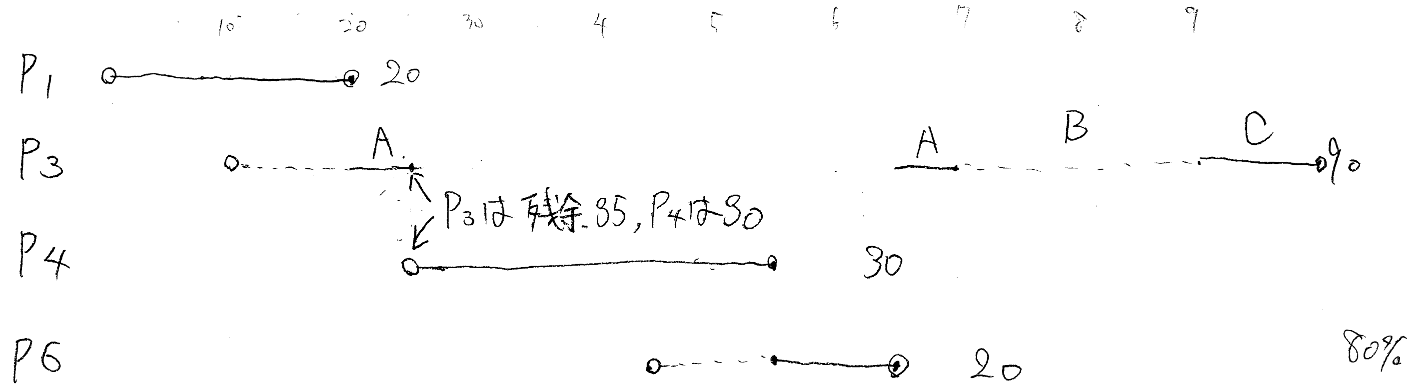
(2-2-1) (2-2-2)

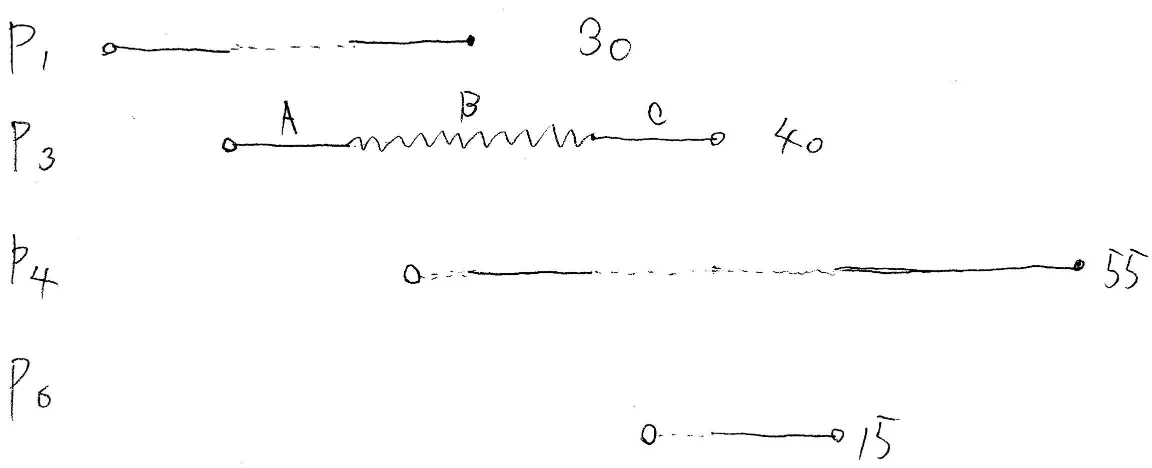
	P1	P2	P4	P5	avg
(a)	20	55	65	60	50
(b)	20	55	75	30	45

	P1	P3	P4	P6	avg	利用率
(c)	20	90	30	20	40	80%
(d)	30	40	55	15	35	100%


0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100
P1
20
P2
55
P4
65
P5
60


20
55
75
90


10 20 30 4 5 6 7 8 9
P1
20
P3
A
A
B
C
90
P3は残余35, P4は30
P4
30
P6
20
80%


P1
30
P3
A
B
C
40
P4
55
P6
15

大問 4 計算理論

(1-1-1)

	a	b	c	d
p	q	p	p	p
q	q	q	q	q

(1-1-2)

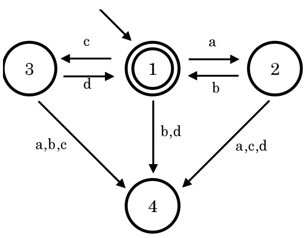
関数 a を一度も呼び出さない引数 x は存在するから．($\overline{L_1}$ は a をひとつも含まない言語)

(1-2-1)

・ 関数が呼び出されなくても受理状態
・ 最初に呼び出されるのは a または c
・ a の次に呼び出されるのは b，かつ，c の次に呼び出されるのは d
・ b または c が呼び出された後は a または c が呼び出されるか，受理される

(1-2-2)

状態遷移図を求め，これをもとに状態遷移表を作成．



	a	b	c	d
→*1	2	4	3	4
2	4	1	4	4
3	4	4	4	1
4	4	4	4	4

(2-1)

2^k

(2-2)

$|v| < k + 1$

根から葉までの経路は最大 k+1，非終端記号も k+1 個．それが非終端記号しの種類数よりも多ければ題意を満たす．

(2-3)

(ア) $2^v - 1$

電子回路と論理設計

回答例

(1)

状態遷移表

	$x=0$	$x=1$
S_0	S_0	S_1
S_1	S_4	S_0
S_2	S_2	S_1
S_3	S_4	S_5
S_4	S_2	S_3
S_5	S_4	S_0

出力表

	$x=0$	$x=1$
S_0	0	0
S_1	1	1
S_2	0	0
S_3	1	1
S_4	0	1
S_5	1	1

(2)

$$y = Q_0 + Q_2 x$$
$$Q_2^+ = Q_1 Q_0 + Q_0 \overline{x}$$
$$Q_1^+ = Q_2 \overline{Q_0}\ \overline{x} + Q_2 \overline{Q_0}$$
$$Q_0^+ = \overline{Q_0} x + Q_1 x$$

(3)

CL0：3
CL1：3
CL2：3
CLY：2

(4)

A：S_2
B：S_5
C：S_3
D：-
E：S_4
F：-

(5)

$$y = \overline{Q_1} Q_0 + Q_0 x + Q_1 \overline{Q_0}$$
$$Q_1^+ = \overline{Q_1} Q_0 \overline{x} + Q_1 Q_0 x + Q_1 \overline{Q_0}\ \overline{x}$$
$$Q_0^+ = Q_1 \overline{Q_0} + \overline{Q_0} x + \overline{Q_1} Q_0 \overline{x}$$

(6)

(2) の HW コスト：59
(5) の HW コスト：52

(7)

状態数を減らすと記憶回路を減らすことができるので記憶回路のハードウェアコストは減少するが、記憶回路への入力の最簡積和形の論理式が複雑になり、組合わせ回路のハードウェアコストは増加してしまうため。

解説

(2)

それぞれのカルノー図は以下のようになる。

y		Q_0x			
		00	01	11	10
	00	0	0	1	1
Q_2Q_1	01	0	0	1	1
	11	d	d	d	d
	10	0	1	1	1

Q_2^+		Q_0x			
		00	01	11	10
	00	0	0	0	1
Q_2Q_1	01	0	0	1	1
	11	d	d	d	d
	10	0	0	0	1

Q_1^+		Q_0x			
		00	01	11	10
	00	0	0	0	0
Q_2Q_1	01	1	0	0	0
	11	d	d	d	d
	10	1	1	0	0

Q_0^+		Q_0x			
		00	01	11	10
	00	0	1	0	0
Q_2Q_1	01	0	1	1	0
	11	d	d	d	d
	10	0	1	0	0

(3)

ドモルガンの法則を使うと、Q_2^+, Q_1^+, Q_0^+, y はそれぞれ以下のように書き換えることができる。

$$\begin{aligned} Q_2^+ &= \overline{\overline{Q_1Q_0}\cdot\overline{Q_0\overline{x}}} \\ Q_1^+ &= \overline{\overline{Q_2\overline{Q_0}\;\overline{x}}\cdot\overline{Q_2\overline{Q_0}}} \\ Q_0^+ &= \overline{\overline{\overline{Q_0}x}\cdot\overline{Q_1x}} \\ y &= \overline{\overline{Q_0}\cdot\overline{Q_2x}} \end{aligned}$$

(5)

それぞれのカルノー図は以下のようになる。

y		x	
		0	1
	00	0	0
Q_1Q_0	01	1	1
	11	0	1
	10	1	1

Q_1^+		x	
		0	1
	00	0	0
Q_1Q_0	01	1	0
	11	0	1
	10	1	0

Q_0^+		x	
		0	1
	00	0	1
Q_1Q_0	01	1	0
	11	0	0
	10	1	1

(6)

ドモルガンの法則を使うと、Q_1^+, Q_0^+, y はそれぞれ以下のように書き換えることができる。

$$
\begin{aligned}
Q_1^+ &= \overline{\overline{\overline{Q_1}Q_0\overline{x}} \cdot \overline{Q_1Q_0x} \cdot \overline{Q_1\overline{Q_0}\ \overline{x}}} \\
Q_0^+ &= \overline{\overline{\overline{Q_1}Q_0\overline{x}} \cdot \overline{Q_1\overline{Q_0}} \cdot \overline{\overline{Q_0}x}} \\
y &= \overline{\overline{\overline{Q_1}Q_0} \cdot \overline{Q_0x} \cdot \overline{Q_1\overline{Q_0}}}
\end{aligned}
$$

※ Q_1^+ と Q_0^+ で $\overline{\overline{Q_1}Q_0\overline{x}}$ が共通している

大問３ 離散構造

（１－１）
α ： D1
β ： A1 と A2

（１－２）
演繹定理

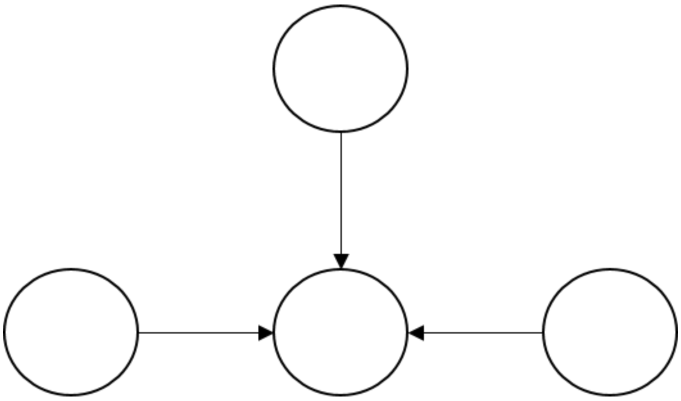
（１－３）
$\Gamma \vdash_D P$、$P \vdash_D Q$、$P, Q \vdash_D R$ を仮定する

$\Gamma \vdash_D = \{P, Q\}$ とすると、$D1$ を用いて
$\Gamma \vdash_D P$
$\Rightarrow P, Q \vdash_D P$
$\Rightarrow P \vdash_D Q \to P$
$\Rightarrow \vdash_D P \to (Q \to P)$
よって $A1$ が成立

$D1$ から、
$P \vdash_D Q \Rightarrow \vdash_D (P \to Q)$
$P, Q \vdash_D R$
$\Rightarrow P \vdash_D (Q \to R)$
$\Rightarrow \vdash_D P \to (Q \to R)$
推論規則 $B1$ から、
$\Gamma \vdash_D P$ かつ $\vdash_D (P \to Q) \Rightarrow \Gamma \vdash_D Q$
$\Gamma \vdash_D Q$ かつ $P \vdash_D (Q \to R) \Rightarrow \Gamma, P \vdash_D R$
$\Gamma = \{P \to (Q \to R)$、$P \to Q\}$ とすると、$D1$ を用いて
$\Gamma, P \vdash_D R$
$\Rightarrow (P \to (Q \to R)), (P \to Q), P \vdash_D R$
$\Rightarrow (P \to (Q \to R)), (P \to Q) \vdash_D (P \to R)$
$\Rightarrow (P \to (Q \to R)) \vdash_D ((P \to Q) \to (P \to R))$
$\Rightarrow \vdash_D ((P \to (Q \to R)) \to ((P \to Q) \to (P \to R)))$
よって $A2$ が成立

（２－１）
$\exists y \forall x\, e(x,y)$ とは, ある頂点はどの頂点からも有向辺があるという意味である。すなわち, どの頂点にも別の頂点へ向かう有向辺があるので、 $\forall x \exists y\, e(x,y)$

（２－２）



（２－３－１）
$A \Leftrightarrow \forall x \exists y\, (\neg Z(x) \wedge Z(y) \wedge e(y,x))$

（２－３－２）
条件から、次の命題が真であることがわかる
$\neg Z(v_2), \neg Z(v_4),\, e(v_1,v_2),\, e(v_2,v_3),\, e(v_3,v_4),\, e(v_4,v_1)$
これらを用いて、$A \to Z(v_1) \wedge Z(v_3)$ が恒真であることを示す
c をスコーレム変数とする
$A \to Z(v_1) \wedge Z(v_3)$
$\Leftrightarrow \neg A \vee (Z(v_1) \wedge Z(v_3))$
$\Leftrightarrow \forall y(Z(c) \vee \neg Z(y) \vee \neg e(y,c) \vee (Z(v_1) \wedge Z(v_3)))$
$\Leftrightarrow \forall y_1(Z(c_1) \vee \neg Z(y_1) \vee \neg e(y_1,c_1) \vee Z(v_1))$
$\quad \wedge \forall y_2(Z(c_2) \vee \neg Z(y_2) \vee \neg e(y_2,c_2) \vee Z(v_3))$
$\Leftrightarrow \forall y_1(Z(v_2) \vee \neg Z(y_1) \vee \neg e(y_1,v_2) \vee Z(v_1))$
$\quad \wedge \forall y_2(Z(v_4) \vee \neg Z(y_2) \vee \neg e(y_2,v_4) \vee Z(v_3))$
$\Leftrightarrow$ 恒真
なぜなら、
$y_1 = v_1$のとき、$\neg Z(v_1) \vee Z(v_1) = true$
$y_1 \neq v_1$のとき、$\neg e(y_1,v_2) = true$
$y_2 = v_3$のとき、$\neg Z(v_3) \vee Z(v_3) = true$
$y_2 \neq v_3$のとき、$\neg e(y_2,v_4) = true$
だから、いかなる $y1$，$y2$ に対しても成立するといえる

（３－１）　真：証明
h が全射
$\Leftrightarrow \forall (y,z) \in Y \times Y\,, \exists x \in X\, s.t.\, h(x) = (f(x), g(x)) = (y,)$
$\Rightarrow \forall y, z \in Y\,, \exists x \in X\, s.t.\, f(x) = y \wedge g(x) = z$
$\Rightarrow \forall y \in Y\,, \exists x_1 \in X\, s.t.\, f(x_1) = y$　かつ　$\forall z \in Y\,, \exists x_2 \in X\, s.t.\, g(x_2) = z$
$\Leftrightarrow f$ が全射かつ g が全射

（３－２）　偽：反例
$X = Y = \{0, 1\}$ とする
$\forall x \in X,\, f(x) = x,\, g(x) = x$ とすると、f と g がともに全射となる
しかし、$h(x) = (f(x)g(x)) = (0, 1) \in Y \times Y$
　を成り立たせる x が存在しないので、h は全射にならないs

（３－３）　偽：反例
$X = Y = \{0, 1, 2\}$
$h(0) = (0, 0) \in Y \times Y$
$h(1) = (0, 1) \in Y \times Y$
$h(2) = (1, 1) \in Y \times Y$
とすると、h が単射となる
しかし、 $f(0) = f(1),\, g(1) = g(2)$ となるため
f と g はともに単射にならない

（３－４）　真： 証明
対偶を示す
h が単射ではない
$\Rightarrow \exists x, y \in X s.t.\, x \neq y$ かつ $h(x) = h(y)$
$\Rightarrow \exists x, y \in X s.t.\, x \neq y$ かつ $(f(x), g(x)) = (f(y), g(y))$
$\Rightarrow \exists x, y \in X s.t.\, x \neq y$ かつ $f(x) = f(y)$ かつ $g(x) = g(y)$
$\Rightarrow f$ が単射でないかつ g が単射でない