平成 25 年度
東京大学大学院情報理工学系研究科
コンピュータ科学専攻
入学試験問題

専門科目 I

平成 24 年 8 月 21 日
10:00 – 12:30

注意事項

(1) 試験開始の合図があるまで, この問題冊子を開けないこと.
Do not open this problem booklet until the start of the examination is announced.

(2) 4 題すべてに答えよ. 問題ごとに指定された解答用紙を使用すること.
Answer the following 4 problems. Use the designated answer sheet for each problem.

(3) 解答用紙および問題冊子は持ち帰らないこと.
Do not take this problem booklet or any answer sheet out of the examination room.

下欄に受験番号を記入すること.
Write your examinee's number in the box below.

受験番号	No.

余白 (blank page)

計算などに使ってもよいが，切り離さないこと． Usable for memos; do not detach.

余白 (blank page)

計算などに使ってもよいが，切り離さないこと． Usable for memos; do not detach.

問題 1

以下では正の整数 n に対し

$$\omega_n = \exp(-2\pi i/n) = \cos(2\pi/n) - i\sin(2\pi/n)$$

とする．また，行列 F_n は (j,k) 要素が

$$\omega_n^{jk} = \overbrace{\omega_n \cdot \omega_n \cdot \cdots \cdot \omega_n}^{jk \text{ times}}$$

であるような $n \times n$ 行列とする．ただし添字は $1 \le j,k \le n$ を動く．I_n は n 次の単位行列，O_n はすべての要素が 0 であるような $n \times n$ の行列とする．行列 A に対して $A^\top$ は A の転置を表す．以下の問いに答えよ．

(1) m が正の整数のとき， $\omega_{4m}^{m}, \omega_{4m}^{2m}, \omega_{4m}^{3m}, \omega_{4m}^{4m}$ を求めよ．

(2) $n \times 2n$ の行列 $J_n = (I_n\ O_n)$ を考える．$\omega_{2n}^{2k} = \omega_n^k$ に注意して，$J_n F_{2n} P_n = F_n$ となるような P_n を求めよ．ただし P_n は $2n \times n$ の行列で，各列はひとつの 1 と $2n-1$ 個の 0 からなる．

(3) P_n の要素をひとつ上にずらしたものを Q_n とする．すなわち

$$(Q_n)_{jk} = \begin{cases} 0 & \text{for } j = 2n \\ (P_n)_{(j+1)k} & \text{otherwise} \end{cases}$$

である．このとき $(P_n\ Q_n)^\top (P_n\ Q_n) = I_{2n}$ となることを示せ．

(4) $\Pi_n = (P_n\ Q_n)$ とする．

$$F_{2n}\Pi_n = \begin{pmatrix} F_n & A_n F_n \\ B_n F_n & C_n F_n \end{pmatrix} = \begin{pmatrix} I_n & A_n \\ B_n & C_n \end{pmatrix} \begin{pmatrix} F_n & O_n \\ O_n & F_n \end{pmatrix}$$

となるような A_n, B_n, C_n を求めよ．また，A_n, B_n, C_n の非零要素数はそれぞれいくつか．

(5) $\boldsymbol{x}$ が $2n$ 次ベクトルのとき，

$$\begin{pmatrix} I_n & A_n \\ B_n & C_n \end{pmatrix} \boldsymbol{x}$$

の計算量を求めよ．

(6) ある正整数 p に対して $n = 2^p$ となるとする．問い (4) の式を再帰的に用い，n 次ベクトル $\boldsymbol{z}$ に対して $F_n\boldsymbol{z}$ を $O(n \log n)$ で計算するアルゴリズムを示せ．

Problem 1

Define ω_n as

$$\omega_n = \exp(-2\pi i/n) = \cos(2\pi/n) - i\sin(2\pi/n)$$

for a positive integer n. Let F_n be an $n \times n$ matrix whose (j,k) element is

$$\omega_n^{jk} = \overbrace{\omega_n \cdot \omega_n \cdot \cdots \cdot \omega_n}^{jk \text{ times}} .$$

Here the indices run for $1 \le j,k \le n$. Let I_n be the $n \times n$ unit matrix, and let O_n be the $n \times n$ matrix whose elements are all zeros. For a matrix A, $A^\top$ represents the transpose of the matrix A. Answer the following questions.

(1) Calculate ω_{4m}^m, ω_{4m}^{2m}, ω_{4m}^{3m}, ω_{4m}^{4m} for a positive integer m.

(2) Let J_n be the $n \times 2n$ matrix defined by $J_n = (I_n\ O_n)$. Find the matrix P_n that gives $J_n F_{2n} P_n = F_n$. Here P_n is a $2n \times n$ matrix, and each column of P_n consists of a single 1 and $2n-1$ zeros. Note the relation $\omega_{2n}^{2k} = \omega_n^k$.

(3) Let Q_n be a matrix defined by shifting P_n upward by one element, that is,

$$(Q_n)_{jk} = \begin{cases} 0 & \text{for } j = 2n, \\ (P_n)_{(j+1)k} & \text{otherwise.} \end{cases}$$

Show $(P_n\ Q_n)^\top (P_n\ Q_n) = I_{2n}$.

(4) Let $\Pi_n = (P_n\ Q_n)$. Find matrices A_n, B_n, and C_n that satisfy

$$F_{2n}\Pi_n = \begin{pmatrix} F_n & A_n F_n \\ B_n F_n & C_n F_n \end{pmatrix} = \begin{pmatrix} I_n & A_n \\ B_n & C_n \end{pmatrix} \begin{pmatrix} F_n & O_n \\ O_n & F_n \end{pmatrix}.$$

Answer, for each of the matrices A_n, B_n, C_n, how many elements of it are non-zeros.

(5) For a vector $\boldsymbol{x}$ of size $2n$, answer the time complexity to compute

$$\begin{pmatrix} I_n & A_n \\ B_n & C_n \end{pmatrix} \boldsymbol{x} .$$

(6) Assume that $n = 2^p$ holds for a positive integer p. Show an algorithm that computes $F_n \boldsymbol{z}$ in time $O(n \log n)$ for a vector $\boldsymbol{z}$ of size n, using the equation of Question (4) recursively.

問題 2

次のようなプログラミング言語を考える．プログラム全体の集合を $\mathcal{P}$ とする．$d_1;\cdots;d_n$ の部分は関数を相互再帰で定義するものとする．

p （プログラム）	$::= d_1;\cdots;d_n;e$	
d （関数定義）	$::= f(x)\ \{e\}$	
e （式）	$::= i$	定数 $i \in \{0,1\}$ を返す．
	$\mid x$	x の値を返す．
	$\mid e_1;e_2$	e_1, e_2 をこの順で評価し，e_2 の値を返す．
	$\mid f(e)$	関数呼び出し．
	$\mid$ `if(*)` $\{e_1\}$ `else` $\{e_2\}$	e_1 又は e_2 を非決定的に選択して評価し，その値を返す．
	$\mid$ `print` e	e を評価し，その値 $i \in \{0,1\}$ を出力し，i を式全体の値として返す．

プログラム p の停止する実行列によって出力されうる有限文字列の集合を「p によって生成される言語」と呼ぶ．停止しない実行列によって生成される文字列は考えないことに注意せよ．例えば次のプログラムによって生成される言語は空である．

$$h(x)\ \{\texttt{print}\ 1; h(x)\};\ h(0)$$

関数呼び出しに値呼び (call-by-value) 戦略を用いた場合に p によって生成される言語を $\mathcal{L}_{\texttt{cbv}}(p)$，名前呼び (call-by-name) 戦略の場合に p によって生成される言語を $\mathcal{L}_{\texttt{cbn}}(p)$ と書く．例えば，以下のプログラム:

$$f(x)\ \{\texttt{if(*)}\ \{x\}\ \texttt{else}\ \{x;x\}\};\ f(\texttt{print}\ 0)$$

を p_0 とすると，値呼びでは関数呼び出し $f(\texttt{print}\ 0)$ において引数が先に評価されるので $\mathcal{L}_{\texttt{cbv}}(p_0) = \{0\}$ であるが，名前呼びでは引数を評価する前に関数が呼ばれるので $\mathcal{L}_{\texttt{cbn}}(p_0) = \{0, 00\}$ である．

以下の問いに答えよ．

(1) 以下のプログラム p_1 について，$\mathcal{L}_{\texttt{cbv}}(p_1)$ および $\mathcal{L}_{\texttt{cbn}}(p_1)$ を示せ．

$$f(x)\ \{\texttt{if(*)}\ \{x\}\ \texttt{else}\ \{x;x\}\};\ f(f(\texttt{print}\ 0))$$

(2) 値呼び戦略で関数呼び出し $f(i)$ の始めから返り値 j で呼び出しが終了するまでの間に出力されうる文字列の集合を $L_{f,i,j}$ とする（ただし $i,j \in \{0,1\}$）．以下によって定義される関数 f_1 を考える．

$$f_1(x)\ \{f_2(f_3(x))\}$$

$L_{f_1,0,1}$ を $L_{f_2,i,j}$ と $L_{f_3,i,j}$ $(i,j \in \{0,1\})$ を用いて表せ．

(3) すべてのプログラム p について，$\mathcal{L}_{\texttt{cbv}}(p)$ は文脈自由言語であることを示せ．（ヒント：問 (2) の各 $L_{f,i,j}$ について非終端記号 $A_{f,i,j}$ を用意し，$\mathcal{L}_{\texttt{cbv}}(p)$ を生成する文脈自由文法を構成せよ．）

(4) $\{\mathcal{L}_{\texttt{cbn}}(p) \mid p \in \mathcal{P}\} \subseteq \{\mathcal{L}_{\texttt{cbv}}(p) \mid p \in \mathcal{P}\}$ は成立するか．成立するならその理由を，成立しないならその反例を示せ．

Problem 2

Consider the following programming language. Let $\mathcal{P}$ be the set of programs. The part $d_1; \cdots ; d_n$ in the program defines functions in a mutually recursive manner.

p (programs)	::=	$d_1; \cdots ; d_n; e$	
d (function definitions)	::=	$f(x)\{e\}$	
e (expressions)	::=	i	return the constant $i \in \{0, 1\}$.
	|	x	return the value of x.
	|	$e_1; e_2$	evaluate e_1, e_2 in this order, and return the value of e_2.
	|	$f(e)$	function call
	|	`if(*)` $\{e_1\}$ `else` $\{e_2\}$	choose e_1 or e_2 in a non-deterministic manner, evaluate it, and return its value.
	|	`print` e	evaluate e, print its value $i \in \{0, 1\}$, and then return i as the value of the whole expression.

We call the set of strings that may be printed by a terminating execution sequence of p *the language generated by* p. Note that we do not consider a string printed by a non-terminating execution sequence. For example, the language generated by the program:

$$h(x)\ \{\texttt{print}\ 1; h(x)\};\ h(0)$$

is empty.

We write $\mathcal{L}_{\texttt{cbv}}(p)$ for the language generated by p in the call-by-value evaluation strategy for function calls, and write $\mathcal{L}_{\texttt{cbn}}(p)$ for the language generated by p in the call-by-name strategy. For example, let p_0 be the following program:

$$f(x)\ \{\texttt{if(*)}\ \{x\}\ \texttt{else}\ \{x; x\}\};\ f(\texttt{print}\ 0).$$

In the call-by-value strategy, the argument is evaluated before the function call $f(\texttt{print}\ 0)$, thus $\mathcal{L}_{\texttt{cbv}}(p_0) = \{0\}$. In the call-by-name, however, the function is called before the argument is evaluated, thus $\mathcal{L}_{\texttt{cbn}}(p_0) = \{0, 00\}$.

Answer the following questions.

(1) For the following program p_1, give $\mathcal{L}_{\texttt{cbv}}(p_1)$ and $\mathcal{L}_{\texttt{cbn}}(p_1)$.

$$f(x)\ \{\texttt{if(*)}\ \{x\}\ \texttt{else}\ \{x; x\}\};\ f(f(\texttt{print}\ 0))$$

(2) Let $L_{f,i,j}(i, j \in \{0, 1\})$ be the set of strings that may be printed from the beginning of a function call $f(i)$ until the end of the call with the return value j, in the call-by-value strategy. Consider the function f_1 defined by:

$$f_1(x)\ \{f_2(f_3(x))\}$$

Express $L_{f_1,0,1}$ in terms of $L_{f_2,i,j}$ and $L_{f_3,i,j}$ $(i, j \in \{0, 1\})$.

(3) Show that $\mathcal{L}_{\texttt{cbv}}(p)$ is a context-free language for every program p. (Hint: For each language $L_{f,i,j}$ in Question (2), prepare a non-terminal symbol $A_{f,i,j}$ and construct a context-free grammar that generates $\mathcal{L}_{\texttt{cbv}}(p)$.)

(4) Does $\{\mathcal{L}_{\texttt{cbn}}(p) \mid p \in \mathcal{P}\} \subseteq \{\mathcal{L}_{\texttt{cbv}}(p) \mid p \in \mathcal{P}\}$ hold? If so, explain the reason. Otherwise, give a counterexample.

問題 3

アルファベット $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$ からなる Σ 文字列 $X \in \Sigma^*$ に対し，以下の符号表に基づいて変換して得られる二進列 ($\{0,1\}^*$ の元) を $g(X)$ と書くものとする．

X	$g(X)$
a	1
b	01
c	10
d	001
e	010
f	100

たとえば，$g(\mathtt{adb}) = 100101$ である．また，二進列 Y に対して，変換の結果が Y に一致するような Σ 文字列の集合を $S(Y)$ と書く．すなわち，

$$S(Y) \;=\; \{X \in \Sigma^* \mid g(X) = Y\} \;.$$

(1) $S(100101)$ を求めよ．

(2) 所与の二進列 Y に対して，$S(Y)$ に属する Σ 文字列 X の長さはすべて等しい．このことを証明せよ．

以下では，各文字 $x \in \Sigma$ についてその出現確率 $p(x)$ がわかっており，それらの文字が互いに独立に出現するような，ランダムな Σ 文字列 $X = x_1 x_2 \ldots x_n$ を考える．これに対し，確率 $P(X)$ を

$$P(x_1 x_2 \ldots x_n) \;:=\; p(x_1)p(x_2)\ldots p(x_n)$$

によって定める．

二進列 Y に対して，ランダムな文字列 X を変換した結果が Y に一致する確率を $P'(Y)$ と書く．すなわち，

$$P'(Y) = \sum_{X \in S(Y)} P(X) \;.$$

(3) 確率 $P'(Y001)$ を，次の確率 (のうちいくつか) を用いて表せ．

$$P'(Y),\; P'(Y0),\; P'(Y1),\; P'(Y00),\; P'(Y01),\; P'(Y10),\; P'(Y11) \qquad (\dagger)$$

ここで $Y001$ は，二進列 Y の後に二進列 001 を連結したものを表す．

(4) 確率 $P'(Y101)$ を，(†) の確率 (のうちいくつか) を用いて表せ．

(5) 長さ m の二進列 Y に対して，上記 $P'(Y)$ を求めるアルゴリズムおよびその時間計算量を述べよ．

(6) 長さ m の二進列 Y に対して，Σ 文字列 $X \in S(Y)$ のうち $P(X)$ が最大である Σ 文字列を求めるアルゴリズムを述べよ (最大のものが複数ある場合はそのうちの任意の 1 つを出力すればよい)． また，その時間計算量を述べよ．

Problem 3

Let $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$ be an alphabet. For a Σ-string $X \in \Sigma^*$, we denote by $g(X)$ the outcome of the transformation according to the following coding table. Obviously $g(x) \in \{0,1\}^*$ is a binary string.

X	$g(X)$
a	1
b	01
c	10
d	001
e	010
f	100

For example, $g(\mathtt{adb}) = 100101$. Given a binary string Y, we denote by $S(Y)$ the set of those Σ-strings which are translated into Y. That is,

$$S(Y) \;=\; \{X \in \Sigma^* \mid g(X) = Y\} \;.$$

(1) What is the set $S(100101)$?

(2) Given a binary string Y, all the Σ-strings that belong to $S(Y)$ have the same length. Prove this fact.

In what follows, we assume that there is a fixed probability $p(x)$ for the occurrence of each letter $x \in \Sigma$, and that those letters occur in a random Σ-string $X = x_1x_2\ldots x_n$ in a mutually independent way. We define the probability $P(X)$ by the following.

$$P(x_1x_2\ldots x_n) \;:=\; p(x_1)p(x_2)\ldots p(x_n)$$

Given a binary string Y, we denote by $P'(Y)$ the probability with which the transformation of a random Σ-string X coincides with Y. That is,

$$P'(Y) = \sum_{X \in S(Y)} P(X) \;.$$

(3) Represent the probability $P'(Y001)$ using (some of) the following probabilities.

$$P'(Y),\; P'(Y0),\; P'(Y1),\; P'(Y00),\; P'(Y01),\; P'(Y10),\; P'(Y11) \qquad (\dagger)$$

Here $Y001$ denotes the binary string Y followed by the binary string 001.

(4) Represent $P'(Y101)$ using (some of) the probabilities (†).

(5) Describe an algorithm that computes $P'(Y)$ for a given binary string Y of length m. Describe its time complexity.

(6) Let Y be a binary string of length m. Describe an algorithm that computes the Σ-string $X \in S(Y)$ that has the largest probability $P(X)$ (if there exist multiple such X, the output can be any one of them). Describe its time complexity.

問題 4

単一 CPU 計算機でページサイズが 4 kbyte のデマンドページングシステムを考える．

(1) 2 つのコード C1，C2 が配列 a を以下のようにアクセスしている．int 型のサイズは 4 byte とし，配列 a は，仮想メモリアドレス 0 番地から a[0][0], a[0][1], a[0][2], ... の順番で配置される．

```
int a[4][1024];
```

コード C1

```
register int   i, j;
for (i = 0; i < 4; i++) {
 for (j = 0; j < 4; j++) {
  a[i][j] = i*j;
 }
}
```

コード C2

```
register int   i, j;
for (j = 0; j < 4; j++) {
 for (i = 0; i < 4; i++) {
  a[i][j] = i*j;
 }
}
```

以下，4 つの仮想ページアクセスパターンを示す．整数値は仮想ページ番号を意味する．C1 および C2 のアクセスパターンはどれか？

A) 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
B) 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7
C) 0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3
D) 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3

(2) プロセス P_1, P_2, P_3 は仮想メモリ空間を共有し，仮想ページ番号 0 から 7 をアクセスする．プロセス実行前にこれら仮想メモリ空間は物理メモリに割り当てられていないとする．その他プロセス実行に必要な仮想メモリ空間は物理メモリに割り当てられ，これらはスワップアウトされない．共有メモリ領域に割り当て可能な空き物理メモリサイズは 16 kbyte (4 ページ分) とし，ページ置換アルゴリズム FIFO を用いてメモリ管理する．以下の問いに答えよ．

(a) 以下の実行順序で時刻 T_1 から時刻 T_7 の間，プロセスがスケジューリングされた時のページフォールトの回数を求めよ．例えば，時刻 T_1 から時刻 T_2 (T_1〜T_2) では，プロセス P_1 が仮想ページ番号 0, 1, 0, 1 の順にアクセスしている．

$T_1 \sim T_2$	$T_2 \sim T_3$	$T_3 \sim T_4$	$T_4 \sim T_5$	$T_5 \sim T_6$	$T_6 \sim T_7$
$P_1\{0, 1, 0, 1\}$	$P_2\{3, 2, 1, 0\}$	$P_3\{0, 3, 2, 3\}$	$P_1\{4, 5, 6, 7\}$	$P_2\{3, 2, 1, 0\}$	$P_3\{0, 1, 2, 3\}$

(b) 以下のように問い (a) とは異なる実行順序でプロセスがスケジューリングされた時のページフォールトの回数を求めよ．

$T_1 \sim T_2$	$T_2 \sim T_3$	$T_3 \sim T_4$	$T_4 \sim T_5$	$T_5 \sim T_6$	$T_6 \sim T_7$
$P_2\{3, 2, 1, 0\}$	$P_3\{0, 3, 2, 3\}$	$P_1\{0, 1, 0, 1\}$	$P_2\{3, 2, 1, 0\}$	$P_3\{0, 1, 2, 3\}$	$P_1\{4, 5, 6, 7\}$

(c) ワーキングセット $\mathrm{WS}(i, j)$ は時刻 T_i から時刻 T_j の間にアクセスされるページ集合を表現し，ワーキングセットサイズ $\mathrm{WSS}(i, j)$ は時刻 T_i から時刻 T_j の間のワーキングセットのサイズとする．例えば，問い (b) において時刻 T_3 から時刻 T_5 のワーキングセットおよびワーキングセットサイズは，$\mathrm{WS}(3, 5) = \{0, 1, 2, 3\}$，$\mathrm{WSS}(3, 5) = 4$ である．ワーキングセット $\mathrm{WS}(i, j)$ およびワーキングセットサイズ $\mathrm{WSS}(i, j)$ を用いて，スケジューリングによってページフォールトの回数に違いが生じる理由を説明せよ．

Problem 4

Consider a demand paging system, whose page size is 4 kbyte, in a single CPU computer.

(1) As shown below, two program codes, C1 and C2, access `int` array `a`. The size of `int` is 4 byte. The array `a` is allocated in the order of `a[0][0]`, `a[0][1]`, `a[0][2]`, ... from virtual memory address 0.

```
int a[4][1024];
```

```
Code C1
register int   i, j;
for (i = 0; i < 4; i++) {
 for (j = 0; j < 4; j++) {
  a[i][j] = i*j;
 }
}
```

```
Code C2
register int   i, j;
for (j = 0; j < 4; j++) {
 for (i = 0; i < 4; i++) {
  a[i][j] = i*j;
 }
}
```

The following lists show four virtual page access patterns. Each integer value represents a virtual page number. Which are the access patterns in C1 and C2, respectively?

A) 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
B) 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7
C) 0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3
D) 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3

(2) Processes P_1, P_2, and P_3 share a virtual memory space. They access virtual page numbers 0 through 7. Before execution of those processes, no physical memory is allocated for the shared memory space. Besides the shared memory space, other virtual memory spaces are allocated in the physical memory areas and never swapped out. Suppose that 16 kbyte (4 pages) of free physical memory can be used for the shared memory space and that the page replacement algorithm is FIFO. Answer the following questions.

(a) Calculate the number of page faults in the case of the following process scheduling from time T_1 to time T_7. In this chart, for example, process P_1 accesses virtual page numbers 0, 1, 0, and 1 in this order from time T_1 to time T_2 ($T_1 \sim T_2$).

$T_1 \sim T_2$	$T_2 \sim T_3$	$T_3 \sim T_4$	$T_4 \sim T_5$	$T_5 \sim T_6$	$T_6 \sim T_7$
$P_1\{0, 1, 0, 1\}$	$P_2\{3, 2, 1, 0\}$	$P_3\{0, 3, 2, 3\}$	$P_1\{4, 5, 6, 7\}$	$P_2\{3, 2, 1, 0\}$	$P_3\{0, 1, 2, 3\}$

(b) Calculate the number of page faults in the case of the following process scheduling, that is different execution order from Question (a).

$T_1 \sim T_2$	$T_2 \sim T_3$	$T_3 \sim T_4$	$T_4 \sim T_5$	$T_5 \sim T_6$	$T_6 \sim T_7$
$P_2\{3, 2, 1, 0\}$	$P_3\{0, 3, 2, 3\}$	$P_1\{0, 1, 0, 1\}$	$P_2\{3, 2, 1, 0\}$	$P_3\{0, 1, 2, 3\}$	$P_1\{4, 5, 6, 7\}$

(c) Let the *working set* $\mathrm{WS}(i, j)$ represent the set of pages accessed from time T_i to time T_j and the *working set size* $\mathrm{WSS}(i, j)$ represent the size of the working set from time T_i to time T_j. For example, the working set and its size from time T_3 to T_5 shown in Question (b) are $\mathrm{WS}(3, 5) = \{0, 1, 2, 3\}$ and $\mathrm{WSS}(3, 5) = 4$, respectively. Explain why the numbers of page faults are different in different scheduling, using the *working set* $\mathrm{WS}(i, j)$ and the *working set size* $\mathrm{WSS}(i, j)$.

余白 (blank page)

計算などに使ってもよいが，切り離さないこと． Usable for memos; do not detach.

余白 (blank page)

計算などに使ってもよいが，切り離さないこと． Usable for memos; do not detach.