

平成23（2011）年度
東京大学大学院学際情報学府学際情報学専攻
修士課程（総合分析情報学コース）
入学試験問題
専門科目

（平成22年8月23日 14：00～16：00）

試験開始の合図があるまで問題冊子を開いてはいけません。開始の合図があるまで、下記の注意事項をよく読んでください。

(Please read the instructions on the backside.)

1. 本冊子は、総合分析情報学コースの受験者のためのものである。
2. 本冊子の本文は16ページである。落丁、乱丁、印刷不鮮明の箇所などがあった場合には申し出ること。
3. 本冊子には、計8問の問題が収録されている。この8問の中から4問を選択して解答すること。
4. 本冊子の問題には、日本語文と英語文があるが、日本語文が正式なもので、英語文はあくまでも参考である。両者に意味の違いがある場合は、日本語文を優先すること。
5. 解答用紙は4枚ある。選択した問題ごとに解答用紙1枚を使用すること。このほかにメモ用紙が1枚ある。なお、解答用紙のみが採点の対象となる。
6. 解答用紙の上方の欄に、選択した問題の番号及び受験番号を必ず記入すること。問題番号及び受験番号を記入していない答案は無効である。
7. 解答には必ず黒色鉛筆（または黒色シャープペンシル）を使用すること。
8. 解答は原則として日本語によるものとする。ただし、英語で解答しても採点の対象とする。
9. 試験開始後は、中途退場を認めない。
10. 本冊子、解答用紙、メモ用紙は持ち帰ってはならない。
11. 次の欄に受験番号と氏名を記入せよ。

受験番号	
氏 名	

総合分析情報学 第1問 (Question A1)

(1) 右手系をなす直行座標系において、ベクトル $\mathbf{A}, \mathbf{B}, \mathbf{C}$ が

$$\mathbf{A} = \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 0 \\ 2 \\ -1 \end{bmatrix}$$

であるとき、次のベクトル積とスカラー3重積を求めよ。

(a) $(\mathbf{A} \times \mathbf{B}) \times \mathbf{C}$ (b) $(\mathbf{C}, \mathbf{A}, \mathbf{B})$

(2) 次の行列 \mathbf{A} が

$$\mathbf{A} = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

であるとき、次を求めよ。

(a) $\det \mathbf{A}$ (b) \mathbf{A}^{-1}

(3) 次の定積分を求めよ。

(a) $\int_0^1 \frac{1}{\sqrt{2-x^2}} dx$ (b) $\int_1^e x(\log_e x)^2 dx$

Question A1

- (1) Consider a rectangular coordinate system (right-hand system) and vector \mathbf{A} , \mathbf{B} , and \mathbf{C} are defined as:

$$\mathbf{A} = \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 0 \\ 2 \\ -1 \end{bmatrix}$$

Calculate the vector product (a) and the scalar triple product (b).

$$(a) \quad (\mathbf{A} \times \mathbf{B}) \times \mathbf{C} \qquad (b) \quad (\mathbf{C}, \mathbf{A}, \mathbf{B})$$

- (2) Assume a matrix \mathbf{A} :

$$\mathbf{A} = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Calculate (a) and (b).

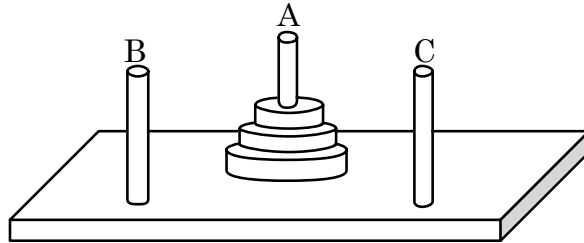
$$(a) \quad \det \mathbf{A} \qquad (b) \quad \mathbf{A}^{-1}$$

- (3) Calculate (a) and (b).

$$(a) \quad \int_0^1 \frac{1}{\sqrt{2-x^2}} dx \qquad (b) \quad \int_1^e x(\log_e x)^2 dx$$

総合分析情報学 第2問 (Question A2)

台の上に3本の棒A、B、Cがある。今Aの棒にN枚の穴の空いた円盤が重ねられている。円盤の大きさはすべて異なり、下にいくほど大きくなっている。ここで、円盤の番号を一番上から、1番、2番、...、N番とする。



問題は、Aの棒にあるN枚の円盤を、次の規則に従ってBの棒に移動することである。

規則1：1回に1枚の円盤しか移動してはならない。

規則2：常に下の円盤が上の円盤より大きくなければならない。

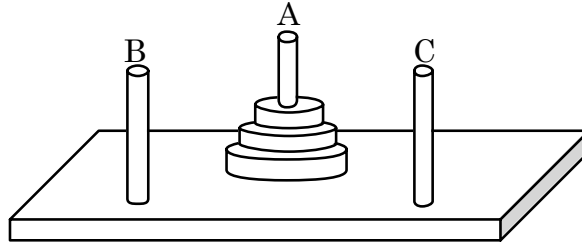
規則3：円盤は3本の棒A、B、C以外に置いてはいけない。

- (1) 円盤の枚数がNのとき、棒Aから棒BへN枚の円盤を移動する手順を出力する再帰的なアルゴリズムを文章で示せ。
- (2) (1)のアルゴリズムをNを引数としてC言語で実装したプログラムを示せ。
- (3) (1)のアルゴリズムに従って棒Aから棒BへN枚の円盤を移動するために必要最小限の円盤の移動回数 $M(N)$ を求めよ。また、 $M(10)$ を計算せよ。
- (4) 以下の再帰的呼び出しを行っているC言語のプログラムを、再帰的呼び出しを使わないプログラムに変形せよ。ここで、Sの部分は1個以上のreturn文を含み関数Fの呼び出しを行わない文の並びである。

```
void F(int x, int y) {  
    int a, b;  
    ... S ...  
    F(a, b);  
    return;  
}
```

Question A2

Consider three rods A, B, and C, and N disks of different diameters that may slide onto any rod. As shown in the figure below, rod A has all the disks in ascending order of diameter. The disks are numbered from top to bottom as No.1, No.2,..., and No.N.



Suppose moving N disks from rod A to rod B observing the following rules.

Rule 1: Only one disk may move at a time.

Rule 2: Disks must be stacked in ascending order of diameter.

Rule 3: Disks must stay on either of three rods, A, B, and C.

- (1) Describe a recursive algorithm for moving N disks from rod A to rod B.
- (2) Implement the algorithm in (1) as a function of N in C language.
- (3) Using the algorithm in (1), obtain the minimum number of moves, $M(N)$, for transferring all the N disks from rod A to B. Also calculate $M(10)$.
- (4) Transform the following recursive program in C language into a program without a recursive call. Here, the part S includes one or more “return” sentences and does not call function F.

```
void F(int x, int y) {  
    int a, b;  
    ... S ...  
    F(a, b);  
    return;  
}
```

総合分析情報学 第3問 (Question A3)

プログラミング言語における以下の問いに答えよ。

- (1) オブジェクト指向言語とは何か説明せよ。
- (2) 以下の用語を簡潔に説明せよ。
 - (a) 多重継承
 - (b) 抽象クラス
 - (c) 純粋仮想関数
 - (d) 仮想関数テーブル (vtable)
 - (e) ダック・タイピング
- (3) 以下のC++プログラムで定義された class A, B, C のインスタンス a, b, c に対するメモリレイアウトを図示し簡潔に説明せよ。
- (4) 以下のC++プログラムで定義された class C のインスタンス c へのポインタを pc とする。pc->va() の呼び出しがどのように行われるか簡潔に説明せよ。

```
class A
{
    public:
        void fa() {}
        virtual void va() {}
        int ia;
}
class B : public A
{
    public:
        void fb() {}
        virtual void vb() {}
        int ib;
}
class C : public B
{
    public:
        void fc()
        virtual void va() {}
        virtual void vb() {}
        int ic;
}
```

Question A3

Answer the following questions regarding programming languages.

- (1) Define object-oriented programming languages.
- (2) Concisely explain the following terminologies.
 - (a) multiple inheritance
 - (b) abstract class
 - (c) pure virtual function
 - (d) virtual method table (vtable)
 - (e) duck typing
- (3) Illustrate with brief explanation the memory layout of instances **a**, **b** and **c** of **class A**, **B** and **C**, respectively, defined in the following C++ program.
- (4) Suppose **pc** be a pointer to instance **c** of **class C** defined in the following C++ program. Briefly explain how **pc->va()** gets called.

```
class A
{
    public:
        void fa() {}
        virtual void va() {}
        int ia;
}
class B : public A
{
    public:
        void fb() {}
        virtual void vb() {}
        int ib;
}

class C : public B
{
    public:
        void fc()
        virtual void va() {}
        virtual void vb() {}
        int ic;
}
```

総合分析情報学 第4問 (Question A4)

- (1) 割り込み処理に関する以下の問いに答えよ。
- (a) 割り込み処理とはどのような処理か説明せよ。
 - (b) サブルーチンと割り込みの処理は、どのような点で異なるか説明せよ。
- (2) パイプラインハザードに関する以下の問いに答えよ。C言語で図1のコードをコンパイルしたところ図2のような機械語に翻訳された。変数のメモリマップは図3を参照せよ。
- (a) CPUがこの機械語をパイプライン処理している場合に、どのようなパイプラインハザードがありうるか説明せよ。
 - (b) 上記のパイプラインハザードを回避するために、図2の機械語を一部修正した機械語を記せ。その際、なぜそのハザードが回避できるのか説明せよ。

```

A = B + C;
D = B + E;
```

図1：C言語による演算コード

```

LOAD   R1 S0(4)
LOAD   R2 S0(8)
ADD    R3 R1 R2
STORE  S0(0) R3
LOAD   R2 S0(16)
ADD    R3 R1 R2
STORE  S0(12) R3
```

図2：図1のコードを翻訳した機械コード

注：

SO (ADR) : (SOレジスタの値 + ADR) 番地のアドレスを表す。

LOAD A B : メモリBのデータをレジスタAにロードする。

STORE A B : レジスタBのデータをメモリAにストアする。

ADD A B C : レジスタBのデータとレジスタCのデータを足してレジスタAに格納する。

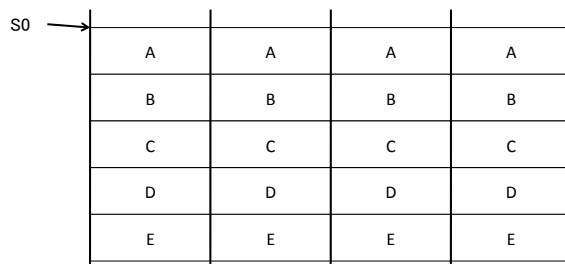


図3：変数のメモリマップ

Question A4

- (1) Answer the following questions on interrupt processing.
 - (a) Define interrupt processing.
 - (b) Explain the difference between interrupt processing and subroutine.

- (2) Answer the following questions regarding pipeline hazard. The code in C language shown in Figure 1 is compiled into machine language as shown in Figure 2. Refer to Figure 3 for the memory map.
 - (a) Explain pipeline hazards that may occur in the pipeline processing of the machine language code in CPU.
 - (b) To avoid the above pipeline hazard, how can we modify the machine language code of Figure 2? Explain why the modified code can avoid the hazard.

<pre>A = B + C; D = B + E;</pre>

Figure 1: The source code in C language

<pre>LOAD R1 S0(4) LOAD R2 S0(8) ADD R3 R1 R2 STORE S0(0) R3 LOAD R2 S0(16) ADD R3 R1 R2 STORE S0(12) R3</pre>
--

Figure 2: The machine language code compiled from the code shown in Figure 1

Note: S0 (ADR) means the address of “S0 register value + ADR”

LOAD A B : load the value of memory B into register A.

STORE A B : store the value of register B into memory A.

ADD A B C : store the sum of data B and data C into register A.

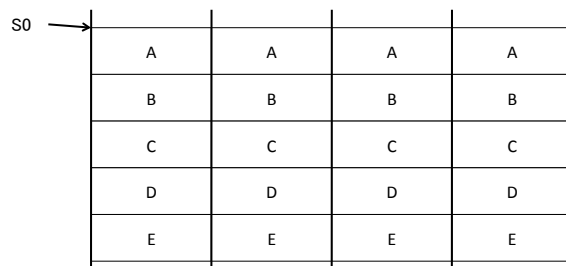


Figure 3: The memory map of the variables

総合分析情報学 第5問 (Question A5)

以下の表に示すプロセスの集合のスケジューリングについて後の問いに答えよ。ここでは、単一プロセッサ上のスケジューリングを仮定し、到着しているプロセスはすべてスケジュールされるものとする。各プロセスの応答時間は、そのプロセスが到着してから終了するまでの時間であり、平均応答時間は、すべてのプロセスの応答時間の和をプロセスの数で割って求める。

プロセス	到着時間	処理時間
A	0	6
B	1	4
C	2	2

- (1) 到着順スケジューリングアルゴリズムを適用したときの平均応答時間を求めよ。プロセスは終了するまで途中で切り替わらないとし、プロセスの切替えに要する時間は無視できるものとする。
- (2) 処理時間順スケジューリングアルゴリズムを適用したときの平均応答時間を求めよ。プロセスは終了するまで途中で切り替わらないとし、プロセスの切替えに要する時間は無視できるものとする。
- (3) プリエンプションのある処理時間順（プロセスが到着した時点で、残りの処理時間が最小のものを選択して実行する手法）を適用したとき、平均応答時間を求めよ。プロセスの切替えに要する時間は無視できるものとする。
- (4) プロセスの切替えに要する時間を s とするとき、(2) と (3) のスケジューリングアルゴリズムの平均応答時間を求めよ。(3) の平均応答時間が (2) の平均応答時間より短くなる条件を s の式で示せ。ただし、 $s < 1$ とする。
- (5) 実際のオペレーティングシステムでは、プロセスの処理時間はあらかじめわからないことが多いため、予想される処理時間をもとに処理時間が短いプロセスを優先するスケジューリングが行われることが多い。そのようなスケジューリングの例を一つ挙げ、その概要を簡単に説明せよ。

Question A5

Suppose that the following processes arrive for execution at times indicated. Each process will run for the amount of time listed. In answering the questions, we assume that all the arrived processes should be scheduled and that the scheduling is performed for a single processor. The response time is the time since the process arrived for execution till the process completes its execution, and the average response time is the total response times divided by the number of processes.

process	arrived time	process time
A	0	6
B	1	4
C	2	2

- (1) What is the average response time for these processes with the first come, first served (first in, first out) scheduling algorithm? We assume that the processes are not preempted until it completes its execution and that the time required for context switch is zero.
- (2) What is the average response time for these processes with the shortest processing time first scheduling algorithm? We assume that the processes are not preempted until it completed its execution and that the time required for context switch is zero.
- (3) When the preemptive shortest processing time first scheduling algorithm is used, what is the average response time for these processes. In the preemptive shortest processing time first scheduling algorithm, when some process arrives, a process with the shortest remaining processing time is scheduled. We assume that the time required for context switch is zero.
- (4) When the context switch time is s , what are the average response times for these processes with the non-preemptive and preemptive scheduling algorithms (2) and (3) ? What is the condition that the average response time for scheduling algorithm (3) is shorter than that for scheduling algorithm (2). We assume $s < 1$.
- (5) In the real operating systems, the processing times are not known in advance and some scheduling algorithms are widely used which gives a short process higher priority based on the estimated processing times of processes. Give an example of such scheduling algorithms that can be implemented on the real operating systems, and explain its outline.

総合分析情報学 第6問 (Question A6)

現在のインターネットにおける経路制御について以下の問いに答えよ。

- (1) ルーティングとフォワーディングについて両者の違いがわかるように説明せよ。
- (2) 最短経路を求める問題を考える。 $d_x(y)$ を x から y への最短経路のコスト, $c(x, v)$ を隣接する x, v 間のリンクのコストとすると、 $d_x(y)$, $d_v(y)$, $c(x, v)$ の間にはある関係が成り立つ。この関係式を求めよ。
- (3) Distance Vector (DV) は、前述の問題で $d_x(y)$ を分散的に求めるアルゴリズムである。 $D_x(y)$ を $d_x(y)$ の近似値と表すとき、DV アルゴリズムを定義せよ。
- (4) 3つのノード x, y, z があり、 $c(x, y) = 4$, $c(y, z) = 1$, $c(z, x) = 50$ とする。また、リンクコストはノードペアに関し対称であると仮定する。すなわち $c(u, v) = c(v, u)$ が成り立つ。ある時刻に $c(x, y) = 1$ となった場合の DV アルゴリズムの挙動を示せ。
- (5) 上記(4)のあと、ある時刻に $c(x, y) = 60$ となった場合の DV アルゴリズムの挙動を示せ。
- (6) 上記(5)における、この挙動を回避するためにはどのような工夫が必要か？
- (7) インターネットの経路制御は階層構造により構成されることを具体的な例を挙げて詳しく説明せよ。

Question A6

Answer the following questions regarding the Internet routing.

- (1) Explain the difference between routing and forwarding.
- (2) Consider the shortest path problem. Suppose $d_x(y)$ be the cost of the shortest path from x to y , $c(x, v)$ be the cost of the link between neighboring nodes x and v , then there holds a formula among $d_x(y)$, $d_v(y)$, and $c(x, v)$. Obtain the formula.
- (3) Distance Vector (DV) is the algorithm for calculating $d_x(y)$ in a distributed manner. Suppose $D_x(y)$ be the estimated value of $d_x(y)$, define DV algorithm.
- (4) Suppose we have three nodes, x , y , and z , and $c(x, y) = 4$, $c(y, z) = 1$, $c(z, x) = 50$. Also a link cost is symmetric in terms of the node pair, that is, $c(u, v) = c(v, u)$ holds. When $c(x, y) = 1$ is enforced, describe how DV behaves to achieve the shortest path.
- (5) After the change in (4), when $c(x, y) = 60$ is enforced, describe how DV behaves to achieve the shortest path.
- (6) How can we avoid the behavior of DV in (5)?
- (7) Explain in detail, giving example(s), that the routing control in the Internet is conducted in a hierarchical structure.

総合分析情報学 第7問 (Question A7)

(1) 次のブール式を簡約化せよ。

(a) $\overline{A}B\overline{C}D + ABC\overline{D} + BCD + \overline{A}BC$

(b) $ABC + BC + A\overline{B}C$

(c) $ABC + \overline{A}B\overline{C} + A\overline{B}C + \overline{A}B\overline{C}$

(2) 組み合わせ回路について以下の問いに答えよ。

(a) 以下の組み合わせ回路をゲート図を用いて示せ。

(i) 2ビット入力で、値が1のビットの数が偶数の時に1を出力する、2ビット偶数パリティチェック回路を設計せよ

(ii) 上記の2ビット偶数パリティチェック回路を用いて、4ビット偶数パリティチェック回路を設計せよ。

(iii) 8ビットの偶数パリティチェック回路を設計せよ。

(b) 2ビットの2進数がある。これに対してその2乗に等しい出力を生ずる組み合わせ回路を設計せよ。

Question A7

(1) Simplify the following boolean functions.

(a) $\overline{A}B\overline{C}D + ABC\overline{D} + BCD + \overline{A}BC$

(b) $ABC + BC + A\overline{B}C$

(c) $ABC + \overline{A}B\overline{C} + A\overline{B}C + \overline{A}B\overline{C}$

(2) Answer the following questions regarding combinational logic circuit.

(a) Show the following logic gate diagrams.

(i) Design a 2-bit even-parity check logic circuit that has 2-bit input, and 1-bit output and that generates the value '1' only when the number of 1's is zero or two.

(ii) Design a 4-bit even-parity check logic circuit by combining the 2-bit even-parity check logic circuits in the previous question.

(iii) Design an 8-bit even-parity check logic circuit.

(b) Design a combinational logic circuit which has input of 2-bit binary number, and output of square of the input number.

総合分析情報学 第8問 (Question A8)

- (1) 空間データ・地理情報に関する以下の (a) から (c) の語句群について、それぞれ簡潔に説明せよ。
 - (a) 空間分布、密度、分析単位 (単位地区)
 - (b) 測量、基準点、電子化
 - (c) ナビゲーション、屋外、屋内

- (2) 空間の情報に関する以下の各問いに答えよ。
 - (a) 位置情報と誤差・精度について簡潔に論ぜよ。
 - (b) 上記のように不確実性を含むデータを表現する方法について簡潔に論ぜよ。
 - (c) 人間が認識あるいは伝達する空間は曖昧であると言われる。これが意味するところを簡潔に説明し、また位置情報システムの応用面に与える示唆を簡潔に論ぜよ。

Question A8

- (1) Explain the following three sets of terms concerning geographic information and spatial data concisely.
 - (a) spatial distributions, density, areal units
 - (b) surveying, control points, digitization
 - (c) navigation, outdoor locations, indoor locations

- (2) Answer the following questions about spatial information.
 - (a) Discuss location information and accuracy/precision concisely.
 - (b) Briefly discuss methods for presenting data that contain uncertainty.
 - (c) Spaces that humans understand or convey are said to be ambiguous. Briefly discuss its meaning, and its implications for the applications of location-based systems.

**Entrance Examination for Masters Program
in Applied Computer Science Course,
Graduate School of Interdisciplinary Information Studies,
The University of Tokyo.
Academic Year 2011
(14:00-16:00, August 23rd, 2010)**

Directions: Do not open this booklet before the examination begins.
Read the following instructions carefully.

1. This booklet is for the examinees in Applied Computer Science Course, Graduate School of Interdisciplinary Information Studies.
2. This booklet includes sixteen pages. Report missing, misplaced, and imperfect pages to the instructor.
3. This booklet includes eight questions. Select any four questions and answer only those four.
4. Each question is described both in Japanese and in English. Use the Japanese version primarily; the English version is provided for the reference purpose only.
5. There are four answer sheets and a scratch paper. Use one answer sheet per question. A scratch paper is provided for calculation. Only the answer sheets will be considered valid.
6. Write a question number and your examinee's number in the designated boxes located at the top of each answer sheet. The answer missing a question number and/or an examinee's number will not be considered valid.
7. Use only black pencils (or black mechanical pencils).
8. Answer the questions in Japanese as a general rule, although you are also allowed to answer in English.
9. Do not leave the room until the examination is finished.
10. Do not take away this booklet, the answer sheets, and the scratch paper.
11. Write your examinee's number and your name in the designated boxes below.

Examinee's Number	
Name	