

## 2011 Summer Entrance Examination

Department of Creative Informatics  
Graduate School of Information Science and Technology  
The University of Tokyo

# Creative Informatics

### Instructions

1. Do not open this brochure until the signal to begin is given.
2. Write your examinee ID below on this cover.
3. Answer three out of the four problems.
4. Three answer sheets are given. Use a separate sheet for each problem. You may use the backside of the sheet.
5. Write down the examinee ID and the problem ID inside the top blanks of each sheet.
6. Do not take out the sheets and this brochure from this room.

Examinee ID \_\_\_\_\_

This is a blank page.

This is a blank page.

## Problem 1

Assume we have  $n$  different products  $G_1, \dots, G_n$  ( $n \geq 2$ ) whose prices are  $p_1, \dots, p_n$  respectively, and choose  $m$  products  $G_{i_1}, \dots, G_{i_m}$  ( $2 \leq m \leq n$ ) from them so that there are no two identical ones. For given two positive integers  $q_{\min}, q_{\max}$  such that  $q_{\min} < q_{\max}$  and  $p_i < q_{\min}$  for each  $i = 1, \dots, n$ , we want to make  $q_{\min} < \sum_{j=1}^m p_{i_j} < q_{\max}$ , that is, the sum of the prices is between  $q_{\min}$  and  $q_{\max}$ , by choosing an appropriate combination of products. The following Algorithm 1 implements the backtracking algorithm that is one of the solutions to this problem. In the descriptions of Algorithm 1,  $\varepsilon$  represents an empty sequence. In the descriptions of the procedure “back( $\langle G_{i_1}, \dots, G_{i_k} \rangle, S_0$ )”, the first argument is a sequence of products consisting of the elements of the product set expected to be the solution eventually and the second argument is a set of products that are candidates to be added to the first argument.  $k$  is the length of the first argument of this “back” invocation. If  $k = 0$ , the first argument is an empty sequence.

**Algorithm 1:** Invoke “back( $\varepsilon, \{G_1, \dots, G_n\}$ )” where the procedure “back” is defined as follows.

**Procedure** back( $\langle G_{i_1}, \dots, G_{i_k} \rangle, S_0$ ):

**Step 1** Let  $S$  be a local variable representing a set of products. Initiate with  $S = S_0$  and proceed to Step 2.

**Step 2** If  $\sum_{j=1}^k p_{i_j} > q_{\min}$ , output  $\{G_{i_1}, \dots, G_{i_k}\}$  and terminate. Otherwise proceed to Step 3.

**Step 3** If  $S$  is empty, output the information saying “no solutions” and terminate if  $k = 0$ , and return to the invoking procedure if  $k > 0$ . If  $S$  is not empty proceed to Step 4.

**Step 4** Choose one element of  $S$ , remove it from  $S$ , and add this element  $G_{i_{k+1}}$  to the end of the sequence  $\langle G_{i_1}, \dots, G_{i_k} \rangle$ . Create the set of  $G' (\in S)$  whose price  $p'$  satisfies “ $p' + \sum_{j=1}^{k+1} p_{i_j} < q_{\max}$ ”. Denote this set by a variable  $S'$  that is different from  $S$ . Invoke “back( $\langle G_{i_1}, \dots, G_{i_k}, G_{i_{k+1}} \rangle, S'$ )” recursively and go back to Step 3.

Then solve the following questions.

(1) Suppose we execute Algorithm 1 for the four products  $G_1, G_2, G_3, G_4$  whose prices are  $p_1 = 1, p_2 = 2, p_3 = 3, p_4 = 4$  respectively and  $q_{\min} = 8, q_{\max} = 10$ . The following sequence represents an example of the arguments of the “back” procedure invocations.

$$(\varepsilon, \{G_1, G_2, G_3, G_4\}) \rightarrow (\langle G_4 \rangle, \{G_1, G_2, G_3\}) \rightarrow \dots$$

Write an example of the arguments of the “back” procedure invocations succeeding to the above ones in the same format in which the execution of a “back” procedure goes back from Step 4 to Step 3 at least one time.

(2) There are some techniques to execute Algorithm 1 efficiently by decreasing the number of invocations of the “back” procedure. One of them is to choose the element  $G_{i_{k+1}}$  of  $S$  whose price is the highest in Step 4. However, in some cases, this technique does not make the number of invocations the smallest. Show an example of such cases by describing the values of

$n, p_i (i = 1, \dots, n), q_{\min}, q_{\max}$  and the arguments of the “back” procedures along the invocations sequence in the same way as (1).

(3) Assume that we invoke “back( $\langle G_{i_1}, \dots, G_{i_k} \rangle, S_0$ )” for a sequence of products  $\langle G_{i_1}, \dots, G_{i_k} \rangle$  and a set of products  $S_0$ , the number of the elements of  $S_0$  is  $l$ , and  $t_l$  is the maximum number of invocations of the “back” procedure during the execution of “back( $\langle G_{i_1}, \dots, G_{i_k} \rangle, S_0$ )” where the invocations include the invocation of “back( $\langle G_{i_1}, \dots, G_{i_k} \rangle, S_0$ )” itself. Then explain the reason why  $t_l = 1 + \sum_{i=0}^{l-1} t_i$  if  $l \geq 1$ .

(4) Describe the maximum number of invocations of the “back” procedures during the execution of Algorithm 1 for  $n$  products  $G_1, \dots, G_n$  ( $n \geq 2$ ) where the invocations include the invocation of “back( $\varepsilon, \{G_1, \dots, G_n\}$ )” at the beginning of the execution.

## Problem 2

(1) Selection of the number of operands in a 2-input arithmetic instruction and selection of the method to access memory are important in designing the format of instructions of a processor. Answer the following questions on the instruction format.

- 1) Describe the pros and cons of 2-operand instruction format and 3-operand instruction format for a 2-input arithmetic instruction.

Example of 2-operand instruction format	ADD	<i>operand 1, operand 2</i>
Example of 3-operand instruction format	ADD	<i>operand 1, operand 2, operand 3</i>

- 2) Describe the pros and cons of an instruction format where an arithmetic instruction has a memory operand and an instruction format where an arithmetic instruction does not have a memory operand, and instead LOAD and STORE instructions are used to access memory. Note that  $r$  denotes a register operand.

Example of an instruction format where an arithmetic instruction has a memory operand	ADD	$r,$	<i>operand to specify memory address</i>
Example of an instruction format where an arithmetic instruction does not have a memory operand and instead LOAD and STORE instructions are used to access memory	LOAD	$r,$	<i>operand to specify memory address</i>
	STORE	$r,$	<i>operand to specify memory address</i>
	ADD	<i>two or three register operands</i>	

(2) Consider a program that calculates the maximum value and the minimum value of 100 16-bit integers. The instruction set must have sufficient kinds of instructions to write this program. Design an instruction format (bit division of an instruction) and instruction set for a processor whose instruction-width is 16-bit and the data-width is 16-bit. Provide a short description of each instruction. The description of an instruction should be a line or two. Note that the instruction set must have subroutine call and return operations.

(3) Using the instruction set designed in (2), write a program that calculates the maximum value and the minimum value of 100 16-bit integers.

(4) Draw a block-diagram of a processor that has the instruction set designed in (2). Then explain the operation of the arithmetic instruction on the processor from the beginning of the instruction execution (just before the instruction fetch from the memory) to the end of execution. Functional blocks such as ALU (Arithmetic Logic Unit) and registers should be used to draw the block-diagram.

This is a blank page.

### Problem 3

(1) As shown in figure 1, four periodical signals  $S_i(t)$  ( $i = 1, 2, 3, 4$ ) with a period of  $4\Delta t$  are denoted as a sequence of digital signals  $\mathbf{w}_i = (w_{i1}, w_{i2}, w_{i3}, w_{i4})^T$  ( $i = 1, 2, 3, 4$ ). Each element  $w_{ik}$  ( $k = 1, 2, 3, 4$ ) takes on values  $+1$  or  $-1$ . Calculate a  $4 \times 4$  matrix  $W$  which consists of each element;

$$W_{ij} = \frac{1}{4} \mathbf{w}_i^T \mathbf{w}_j \quad (i, j = 1, 2, 3, 4)$$

by using  $\mathbf{w}_i$  shown in figure 1. T denotes transpose.

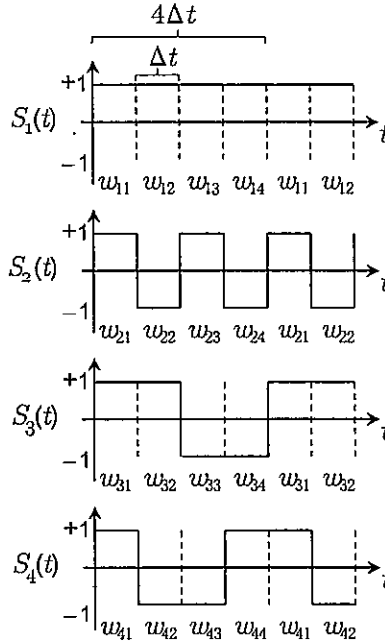


Figure 1

(2) Consider a method of communication shown in figure 2, where a signal period for 1 bit is  $4\Delta t$ . The transmitting signal is generated by multiplying the original signal  $v_i(t)$  ( $i = 1, 2, 3, 4$ ) by the signal  $S_i(t)$  defined in (1). The demodulated signal is generated by multiplying the receiving signal by the signal  $S_i(t)$ . Suppose that the synchronization between the original signal  $v_i(t)$  and the signal  $S_i(t)$  and the synchronization between the receiving signal and the signal  $S_i(t)$  are realized by using an appropriate method. In addition, suppose that there is no delay in the communication line and no effect of noise and attenuation of signals. Answer the following problems.

- 1) Show that the demodulated signal matches to the original signal.
- 2) When the transmitting signal generated by multiplying the original signal  $v_i(t)$  by the signal  $S_i(t)$  is received by the method, write the signal obtained by multiplying the receiving signal by the signal  $S_j(t)$  ( $i \neq j$ ).



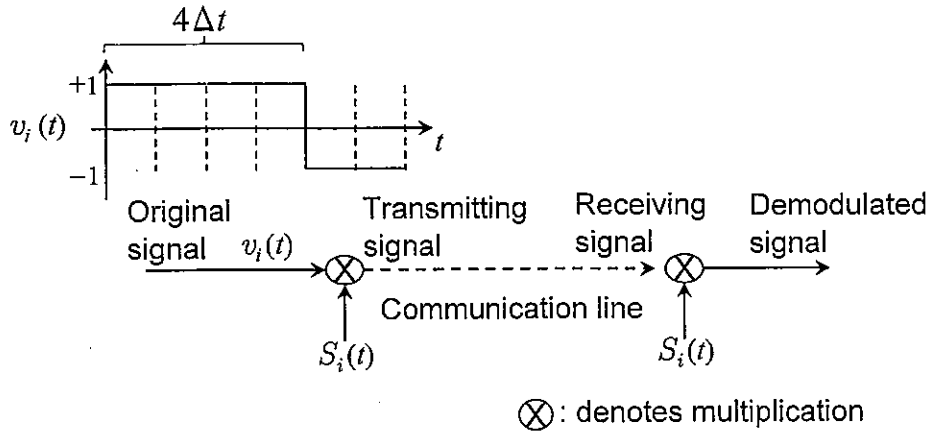


Figure 2

(3) Consider an optical sensing system using concurrently four light sources shown in figure 3. If the same constant lighting levels are used for the light sources, it is difficult to separate the light sources at the photo detectors, so that concurrent sensing cannot be realized. Therefore consider a method for using concurrently four light sources by separating only the signal of the  $i$ -th light source at the  $i$ -th photo detector by using an appropriate signal for the  $i$ -th light source ( $i = 1, 2, 3, 4$ ).

- 1) Show a detailed method for separating the signal of  $i$ -th light source at the  $i$ -th photo detector from the signal of other light sources. Suppose that the synchronizations at the modulation and the demodulation are realized by using an appropriate method. In addition, suppose that there is no delay in the communication line and no effect of noise and attenuation of signals.
- 2) If there is attenuation of the signals in the above situation, show a detailed method for separating the signal of each light source.

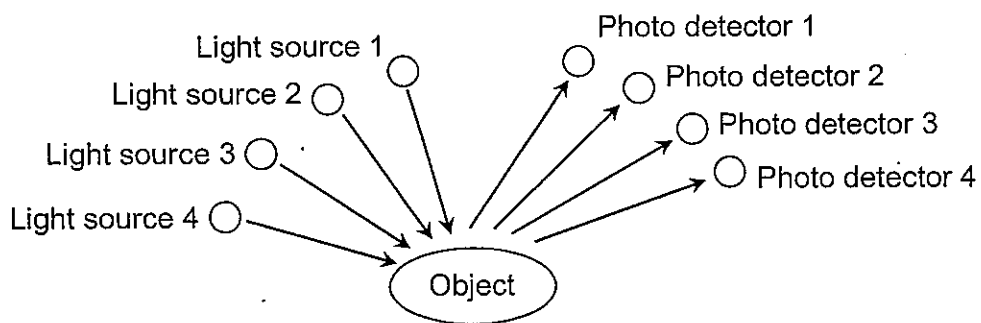


Figure 3

## Problem 4

Select four items out of the following eight items concerning information systems, and explain each item in approximately 4~8 lines of text. If necessary, use examples or images.

- (1) Higher-order function
- (2) PKI
- (3) AS number within the internet
- (4) Technology for localization and mapping in a mobile robot
- (5) Representation of link mechanism in an articulated robot
- (6) Generalized inverse matrix or pseudo-inverse matrix
- (7)  $H^\infty$  control
- (8) Principal component analysis

This is a blank page.

This is a blank page.

This is a blank page.

