2019 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 number below on this cover page.
3. Answer all three problems.
4. Three answer sheets are given. Use a separate sheet of paper for each problem. You may write on the back of the sheet.
5. Write your examinee ID number and the problem number inside the top blanks of each sheet.
6. Do not bring the answer sheets or this brochure out of this room.

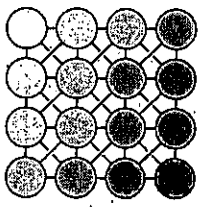
Examinee ID _____

This page is blank.

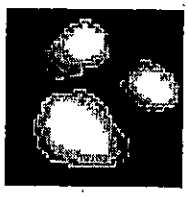
This page is blank.

Problem 1

Consider a 256-level 2-dimensional gray-scale image with $n \times n$ points (pixels). Assume that each point is connected to vertical, horizontal, and diagonal neighbors as shown on the right. We represent each pixel p using a type `Pixel` and its brightness as `p.brightness`. An image is given as an $n \times n$ array `P` of `Pixels`. You can use basic data structures in a pseudo-code. Computational complexity should be given as a function of n.



(1) Assume that we have multiple white objects in a black background as shown on the right. We consider the method of computing the area of one of the white objects as follows.



"We keep points that are brighter than a given threshold and ignore the rest. We then pick a point from the remaining points and compute the size (number of points) of the connected region containing the point."

Give a pseudo-code (equal or less than 20 lines) of an algorithm that executes the computation using recursion and answer its computational complexity using the big-O notation.

(2) We consider the problem of detecting a black curve in a white background (as shown on the right) as follows. Assume that there is no self-intersection.

"Consider a connected point sequence that connects two end points (assume that the end points are given). Among such point sequences, we would like to obtain one with the minimum total brightness (sum of point brightness on a sequence) along this sequence."

Give a pseudo-code (equal or less than 20 lines) of an algorithm that executes the computation efficiently and answer its computational complexity using the big-O notation.

(3) We consider the problem of dividing an image into left and right at a point sequence (as shown on the right) as follows.

"Consider a point sequence that connects top and bottom of the image, containing a single point in each row. Among such point sequences, we would like to obtain one with the minimum total brightness."

Give a pseudo-code (equal or less than 20 lines) of an algorithm that executes the computation efficiently and answer its computational complexity using the big-O notation.

(4) We blur an image by applying the following operation to the image.

"For each internal point (a point that has 8 neighbors), we compute the average of brightness of its 8 neighbors. Once we have computed this average for all the internal points, we update the brightness of each internal point to the corresponding average."

We define three vector representations, x, x', and b where x is a vector listing the

original brightness of internal points, $\mathbf{x}'$ is a vector listing the updated brightness of the internal points, and $\mathbf{b}$ is a vector listing the brightness of the external points (the points of the image other than the internal points). We want to represent the relationship between $\mathbf{x}$, $\mathbf{x}'$, and $\mathbf{b}$ using matrices. Define matrices appropriately and give an equation that describes the relationship using the matrices.

(5) The brightness of the points $\mathbf{x}$ converges to $\mathbf{x}^{\mathrm{inf}}$ after applying the operation defined in (4) for an infinite number of times. Write down an analytic formula for $\mathbf{x}^{\mathrm{inf}}$ using the matrices defined in (4). Do not use limit in the formula.

Problem 2

Let us consider a solar power generation system. Assume that we have operational rules to maintain solar panels as follows; (i) A set of n panels are maintained at the same time as a group. (ii) Each group of panels is periodically examined. (iii) For each group, status of panels is reported as n bit data, where each bit is set to 1 if the corresponding panel is malfunctioning, and 0 otherwise. Consider the "population count" problem where we count the total number of malfunctioning panels, i.e., the number k of 1s in the n bit data. Answer the following questions.

First, let us consider a software solution. Here, $0 < n \leq 32$, $0 \leq k < \log_2 n$. Assume that an arithmetic operation, a logical operation, a shift operation, and a table lookup takes 1 unit time. For simplicity, assume that increments of indices and comparisons for loops take zero time.

(1) A naive method is to check the value of each bit and compute the total sum of the number of 1s. Write down a pseudo-code of this method and answer its computation time.

(2) You can actually improve the computation time of the method (1) via table lookups. Answer its computation time.

(3) Write down a pseudo-code of a method which is faster than the method (1) and requires less storage than the method (2). Answer its computation time.

Let us consider a hardware solution. Here, input is a bit sequence and output is a binary number.

(4) Write down the truth table of a population count logic circuit P_3, where the input is 3 bit. Design P_3 using AND, OR, and NOT gates.

(5) Using logic circuits P_3, design a population count logic circuit P_6, where the input is 6 bit. You may also use additional AND, OR, and NOT gates, if needed.

(6) To design a population count logic circuit P_n, where the input is n bit, latency becomes an issue as n increases. Answer a solution of this latency problem.

This page is blank.

Problem 3

Select four items out of the following eight items concerning information systems, and explain each item in approximately from four to eight lines of text. If necessary, use examples or figures.

(1) Inverse kinematics
(2) Hidden Markov model
(3) MinMax algorithm
(4) NP complete problem
(5) Ray tracing
(6) SIMD (Single Instruction Multiple Data)
(7) Call by value and call by reference
(8) Public-key cryptography

This page is blank.

This page is blank.

This page is blank.