

2020年度夏入試 / 2020 Summer Entrance Examination

東京大学情報理工学系研究科

創造情報学専攻

創造情報学

Department of Creative
Informatics

Graduate School of Information
Science and Technology
The University of Tokyo

Creative Informatics

注意事項

1. 試験開始の合図まで、この問題冊子を開かないこと。
2. この表紙の下部にある受験番号欄に受験番号を記入すること。
3. 3問全てに、日本語ないし英語で解答すること。
4. 解答用紙は3枚配られる。1問ごとに必ず1枚の解答用紙を使用すること。
解答用紙のおもて面に書ききれないときには、うら面にわたってもよい。
5. 解答用紙の指定された箇所に、受験番号および問題番号を忘れずに記入すること。
6. 解答用紙および問題冊子は持ち帰らないこと。

INSTRUCTIONS

1. Do not open this booklet until the start of the examination is announced.
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 answer sheet 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 booklet out of this room.

受験番号 / Examinee ID _____

このページは空白。
This page is blank.

このページは空白。
This page is blank.

第1問

点列 v_0, \dots, v_{n-1}, v_0 をこの順に結んでできる凸 n 角形が与えられたとき、その凸 n 角形の三角形分割とは、その内部を重なりなく三角形に分割する方法のことである。

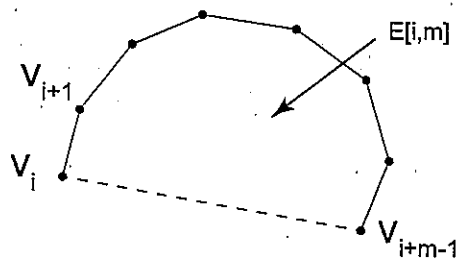
まずは、凸 n 角形の三角形分割の数を求める。その数を $C[n]$ と書く。例えば、 $C[4]=2$ である。

- (1) $C[5], C[6], C[7]$ の値を答えよ。
- (2) $C[n]$ を $C[2], C[3], C[4], \dots, C[n-1]$ の各式を用いて書き表せ。ただし、 $C[2]=1, C[3]=1$ とする。
- (3) 任意の n に対する $C[n]$ を求めるアルゴリズムは、以下のような疑似コードで表現できる。[①]に当てはまるコードを答えよ。またこのアルゴリズムの計算量(オーダー)を答えよ。

```
C[2] = 1; C[3] = 1;
for(i=4...n)
  C[i] = 0;
for(i=4...n)
  for(j=0...i-3)
    [ ① ]
return C[n];
```

次に、凸 n 角形のコスト最小三角形分割を求める。ここで、三角形分割のコストとは、構成する三角形のコストの和であるとし、三角形のコストとはその三角形を構成する辺のコストの和であるとする。また、凸 n 角形を構成する任意の2頂点 v_i と v_j を結ぶ辺のコスト $D[i, j] (=D[j, i])$ はすべて与えられているものとする。

- (4) 与えられた凸 n 角形のコスト最小三角形分割を求める問題を、小問題に分けて解いていくことを考える。頂点 v_i から時計回りに m 個の頂点を訪問し、 v_i に戻ってくる経路によって囲まれる多角形について、コスト最小の三角形分割のコストを $E[i, m]$ と書くことにする(下図参照)。ここで、 $E[i, m']$ が $i=0, \dots, n-1, m'=2, \dots, m-1$ についてすべて計算済みであるとして、 $E[i, m]$ を $E[i, m']$ と $D[i, j]$ の式で表せ。ただし、 $E[i, 2]=0$ ($i=0, \dots, n-1$) とする。またその状況を説明する図も付して示せ。



- (5) 上記 (4) で得られた関係式を用いて、任意の凸 n 角形の三角形分割の最小コストを求めるアルゴリズムの疑似コードを、10行程度で示せ。またその計算量(オーダー)を答えよ。

第2問

ランダム・アクセス可能なメモリを D-FF (Flip Flop) と 2:1 マルチプレクサを用いて作ることを考える。ここで D-FF とは下記の図1に示される1ビットの情報を記憶する回路である。この回路ではクロック信号 clk の立ち上がり時に d に入力されていた1ビットの信号が書き込まれ、それを q に出力し続ける。また、2:1 マルチプレクサとは、下記の図2に示すように選択信号 s に応じて2つの入力信号 a と b どちらか1つの入力信号を選択し c に出力する回路のことである。

- (1) 図2に示される 2:1 マルチプレクサの真理値表を書け。なお、このマルチプレクサでは s が 0 の時に a が選択され、 s が 1 の場合は b が選択されるものとする。
- (2) 4ビットを記憶し、2ビットのアドレスにより指定した位置の1ビットを出力するメモリの回路図を以下の指示に従って描け。
 - ・図1の D-FF と図2の 2:1 マルチプレクサの2種類みを使用すること。
 - ・クロックと書き込みに関わる回路は無視すること (clk と d には何も接続しない)。
 - ・各 D-FF はどのアドレスのデータを記憶しているのかを図中に明記すること。
 - ・2ビットのアドレス信号線のうち下位を $addr_low$ 、上位を $addr_high$ として図中に明記すること。
 - ・メモリの出力信号線を $output$ として図中に明記すること。
- (3) 上記 (2) と同様にして図1の D-FF と図2の 2:1 マルチプレクサを用いて 2^n ビットを記憶し n ビットのアドレスにより指定した位置の1ビットを出力するメモリを作ることを考える。このメモリを作るためには、何個のマルチプレクサが必要になるかを求めよ。
- (4) 下記の図3に示される D-FF は書き込み制御つき D-FF である。この D-FF では、クロック信号 clk の立ち上がり時に、書き込み有効信号 we への入力が 1 であった場合にのみ d から入力された信号が書き込まれる。もし we が 0 であった場合は、書き込みを行わずに以前に書き込まれた信号を q に出力し続ける。この図3の D-FF を、下記図1の D-FF と図2の 2:1 マルチプレクサのみを使って実現し、その回路図を描け。なお、このマルチプレクサでは s が 0 の時に

a が選択され、 s が 1 の場合は b が選択されるものとする。

- (5) 4 ビットを記憶するメモリについて、下記の指示に従ってその回路図を描け。
- ・このメモリではクロック信号の立ち上がり時に、2 ビットのアドレスにより指定した位置に 1 ビットのデータを書き込むことができるものとする。
 - ・図 3 の書き込み制御つき D-FF と図 4 の AND ゲート、図 5 の NOT ゲートのみを使用すること。
 - ・AND ゲートは最大 4 つまで、NOT ゲートは最大 2 つまで使用できるものとする。
 - ・クロック出力に関わる回路は無視してよい (clk と q は何も接続しないでもいい)。
 - ・各 D-FF はどのアドレスのデータを記憶しているのかを図中に明記すること。
 - ・2 ビットのアドレス信号線のうち下位を $addr_low$ 、上位を $addr_high$ として図中に明記すること。
 - ・メモリのデータ入力信号線を $input$ として明記すること。

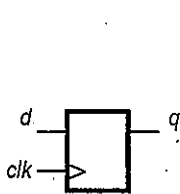


図1 D-FF

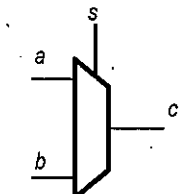


図2 2:1 multiplexer

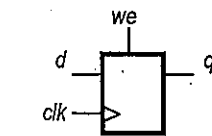


図3 D-FF with write control

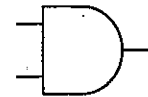


図4 AND gate

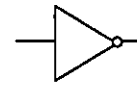


図5 NOT gate

第3問

以下に示す情報システムに関する8項目から4項目を選択し、各項目4～8行程度で説明せよ。必要に応じて例や図を用いて良い。

- (1) セマフォ
- (2) A*探索アルゴリズム
- (3) FPGA
- (4) バッファオーバーフロー
- (5) LR構文解析
- (6) IPv4とIPv6
- (7) ステッピングモータ
- (8) パーセプトロン

このページは空白。
This page is blank.

Problem 1

Given a convex n -gon made by connecting a point sequence v_0, \dots, v_{n-1}, v_0 in this order, a triangulation of the convex n -gon is a way of dividing its interior into triangles without overlap.

First, we count the number of triangulations of a convex n -gon. We denote the number as $C[n]$: For example, $C[4]=2$.

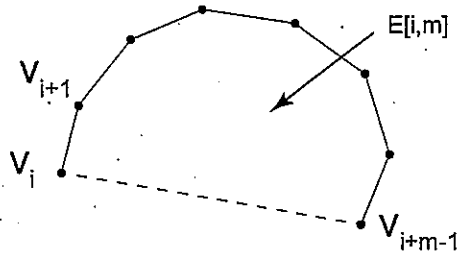
- (1) Answer the values of $C[5]$, $C[6]$, and $C[7]$.
- (2) Represent $C[n]$ as a function of $C[2]$, $C[3]$, $C[4]$, ..., and $C[n-1]$. We define $C[2]=1$ and $C[3]=1$.
- (3) The following pseudo-code implements an algorithm for computing $C[n]$ for arbitrary n . Answer the code that fills [①]. Also answer the computational complexity (order) of the algorithm.

```
C[2] = 1; C[3] = 1;
for(i=4...n)
  C[i] = 0;
for(i=4...n)
  for(j=0...i-3)
    [ ① ]
return C[n];
```

Next, we find the triangulation of a convex n -gon with a minimum cost. Here, the cost of a triangulation is defined as the sum of the cost of triangles in the triangulation, and the cost of a triangle is defined as the sum of the cost of the edges composing the triangle. Assume that all $D[i,j]$ ($=D[j,i]$), the costs of the edges connecting an arbitrary pair of vertices (v_i, v_j) of the n -gon, are given.

- (4) We consider solving the problem of finding a triangulation with a minimum cost by dividing the problem into subproblems. We denote $E[i,m]$ as the cost of triangulating a polygon made by a path starting from v_i , visiting m vertices clockwise, and then coming back to v_i (see the figure below). Assume that $E[i',m']$ are all given for $i'=0, \dots, n-1$ and $m'=2, \dots, m-1$. Represent $E[i, m]$ as a function of $E[i', m']$ and $D[i, j]$. We define $E[i, 2]=0$

($i=0, \dots, n-1$). Also draw a figure explaining the situation.



- (5) Give approximately 10-line pseudo-code implementing an algorithm to compute the minimum cost of triangulating an arbitrary n -gon using the formula obtained in (4). Also answer the computational complexity (order) of the algorithm.

Problem 2

Consider making a memory that can be accessed randomly, using D-FFs (Flip Flop) and 2:1 multiplexers. Assume that the D-FF is a circuit that stores 1 bit as shown in Fig. 1. A 1-bit signal given to d is written to this circuit at the rise of the clock signal clk and this circuit continues to output the written signal to q . As shown in Fig. 2, the 2:1 multiplexer is a circuit that selects one of the two input signals a and b according to the selection signal s and outputs it to c .

- (1) Give a truth table for the 2:1 multiplexer shown in Fig. 2. Assume that this multiplexer selects a when s is 0 and selects b when s is 1.
- (2) Draw a circuit diagram of a memory that stores 4 bits and outputs 1 bit at the position specified by a 2-bit address according to the following instructions.
 - Use only two kinds of components, which are the D-FF shown in Fig. 1 and the 2:1 multiplexer in Fig. 2.
 - Ignore circuits related to clock and write (do not connect anything to clk and d).
 - Specify the address of data stored in each D-FF in the circuit diagram.
 - Specify the lower bit of the 2-bit address signal wires as $addr_low$ and the upper bit of it as $addr_high$ in the circuit diagram.
 - Specify the output wire of the memory as $output$ in the circuit diagram.
- (3) In the similar way as in (2), consider a memory that stores 2^n bits and outputs 1 bit at the position specified by an n -bit address using the D-FF shown in Fig. 1 and the 2:1 multiplexer shown in Fig. 2. Give how many multiplexers you need to make this memory.
- (4) The D-FF shown in Fig. 3 is a D-FF with write control. In this D-FF, the signal given to d is written only when the input to the write enable signal we is 1 at the rise of the clock signal clk . If we is 0, a previously written signal continues to be output to q without updating the stored contents. Give a circuit diagram of this D-FF in Fig. 3 using only the D-FF in Fig. 1 and the 2:1 multiplexer in Fig. 2. Assume that this multiplexer selects a when s is 0 and selects b when s is 1.
- (5) Draw a circuit diagram of a memory that stores 4 bits according to the following instructions.
 - Assume that, at the rise of the clock signal, 1 bit data is written at the position specified by a 2-bit address.
 - Use only three kinds of components, which are the D-FF with write control shown in Fig. 3, the AND gate shown in Fig. 4, and the NOT gate shown in Fig. 5.
 - You can use up to four AND gates and up to two NOT gates.

- Ignore circuits related to clock and output (do not connect anything to *clk* and *q*).
- Specify the address of data stored in each D-FF in the circuit diagram.
- Specify the lower bit of the 2-bit address signal wires as *addr_low* and the upper bit of it as *addr_high* in the circuit diagram.
- Specify the data input wire of the memory as *input* in the circuit diagram.

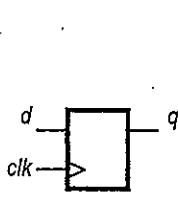


Fig. 1 D-FF

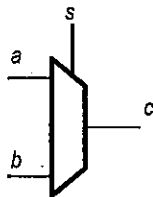


Fig. 2 2:1 multiplexer

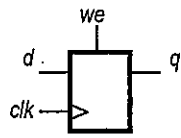


Fig. 3 D-FF with write control

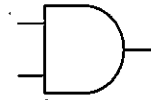


Fig. 4 AND gate

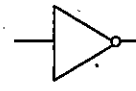


Fig. 5 NOT gate

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) Semaphore
- (2) A* search algorithm
- (3) FPGA
- (4) Buffer overflow
- (5) LR parsing
- (6) IPv4 and IPv6
- (7) Stepping motor
- (8) Perceptron

このページは空白。
This page is blank.

このページは空白。
This page is blank.

このページは空白。
This page is blank.

