平成21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20年8月26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 the answer sheets and the problem booklet out of the examination room.

下欄に受験番号を記入すること.
Fill the following blank with your examinee's number.

受験番号	No.

問題 1

作図法により定義される 2 次元空間の曲線について考える．なお，以下の問いでは 4 点 P_0, P_1, P_2, P_3 はすべて異なり，どの 3 点も同一直線上にはないものとせよ．以下の問いに答えよ．

(1) 図 1 に示すように 3 つの制御点 P_0, P_1, P_2 を考え，それらを結ぶ線分を考える．t を $0 \leq t \leq 1$ を満たす実数として，線分 P_0P_1 を $t : 1-t$ で内分する点を P'_0, 線分 P_1P_2 を $t : 1-t$ で内分する点を P'_1 とする．次に $P'_0P'_1$ を $t : 1-t$ で内分する点を P とする．P が下記で表わされることを示せ．

$$P(t) = (1-t)^2 P_0 + 2(1-t)tP_1 + t^2 P_2$$

(2) t を 0 から 1 まで変化させると点 P は 2 次曲線を描く．これが楕円の一部か，双曲線か，放物線か，いずれであるかを答えよ．

(3) さらに制御点に P_3 を加えて，問い (1) と同じように隣接する点の内分点を求めることを繰り返して P を定める（図 2）．$P(t)$ を問い (1) と同じように P_i に t の多項式を乗じたものの和で表現せよ．

(4) 問い (3) で求めた制御点 4 点の 3 次曲線で，半径 r の四分円を近似したい．図 3 のように点を配置した．$P(1/2) = (r\cos(\pi/4), r\sin(\pi/4))$ となるように P_1, P_2 を定めよ．

(5) 問い (4) の 3 次曲線を $t = 1/2$ で 2 分割する．もとの曲線とまったく同じ曲線になるために，左側および右側の曲線を定義する 2 組 8 つの制御点の位置を求めよ．作図で算出してもよい．

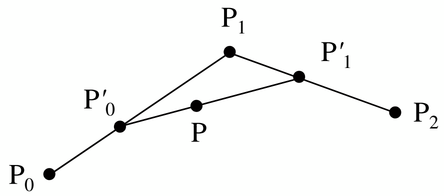


図 1

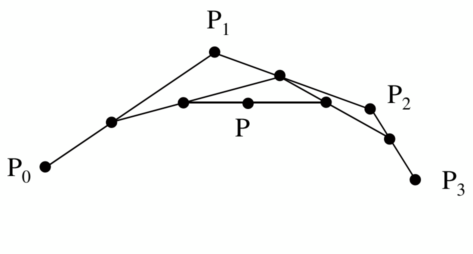


図 2

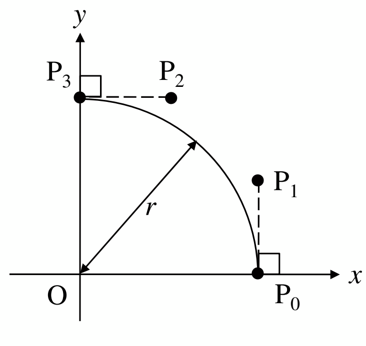


図 3

Problem 1

Consider curves in a two dimensional space defined by figure construction. In the following questions, assume that four points P_0, P_1, P_2, and P_3 are all different and any three points of them are not on a line. Answer the following questions.

(1) Consider three control points P_0, P_1, P_2, and line segments connecting them as in Figure 1. Let t be a real number satisfying $0 \le t \le 1$. Let P_0' divide line segment P_0P_1 in the ratio $t : 1-t$, and, likewise, P_1' divide line segment P_1P_2 in the ratio $t : 1-t$. Further, let P divide line segment $P_0'P_1'$ in the ratio $t : 1-t$. Show that P is expressed as:

$$P(t) = (1-t)^2P_0 + 2(1-t)tP_1 + t^2P_2$$

(2) By moving t from 0 to 1, the point P forms a quadric curve. Is this curve a part of ellipse, a hyperbola, or a parabola?

(3) Add one more control point P_3, and determine P in the same way as question (1), that is, by repeating interior divisions between neighboring points (Figure 2). Express $P(t)$ as the sum of P_i's each multiplied by a polynomial of t, as in question (1).

(4) Consider approximating a quarter of circle of radius r by a cubic curve with four control points as is obtained in question (3). Points are arranged as in Figure 3. Determine P_1 and P_2 so that $P(1/2) = (r\cos(\pi/4), r\sin(\pi/4))$ is satisfied.

(5) Divide the cubic curve in question (4) into two parts at $t = 1/2$. Determine the positions of the two sets of control points that determine the left and the right curves, so that the divided curves are exactly the same as the original curve. You may as well determine them by construction of figures.

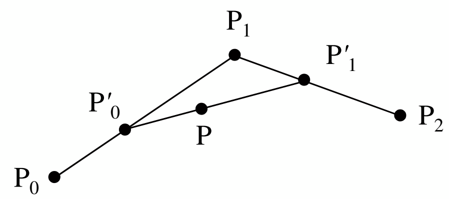


Figure 1

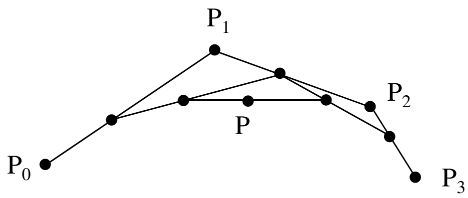


Figure 2

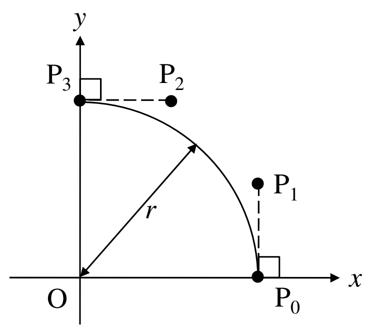


Figure 3

問題 2

A を n 個の要素からなる集合とし，A の部分集合からなる集合 $\Sigma = \{S_i\}_{i=1}^{m}$ は

$$\begin{aligned} S_i \cap S_j &= \emptyset \qquad (i \neq j) \\ \bigcup_{i=1}^{m} S_i &= A \end{aligned}$$

を満たすとする．このとき，Σ に関する以下の操作を考える．

- *member*(S, x): 部分集合 $S \in \Sigma$ に $x \in A$ が含まれているかを調べる．
- *find*(x): $x \in A$ の属している部分集合 $S \in \Sigma$ を求める．
- *merge*(S_i, S_j): $S_i \in \Sigma$ と $S_j \in \Sigma$ を Σ から削除し，$S_i \cup S_j$ を Σ に加える．

以下の問いに答えよ．なお，Σ の要素数は最大で M とせよ．

(1) ビットベクトルによってそれぞれの部分集合を表現することを考える．必要なメモリ空間の大きさの見積もりを与えよ．また，*member*, *find*, *merge* の操作に平均的にどのくらいの手間がかかるのか説明せよ．一般的にビットベクトルによる集合の表現はどのようなときに使うべきであるか説明せよ．

(2) 連結リストによってそれぞれの部分集合を表現することを考える．必要なメモリ空間の大きさの見積もりを与えよ．また，*member*, *find*, *merge* の操作に平均的にどのくらいの手間がかかるのか説明せよ．一般的に連結リストによる集合の表現はどのようなときに使うべきであるか説明せよ．

(3) 要素をキー，その要素が属する部分集合を値とするハッシュ表により集合を表現することを考える．必要なメモリ空間の大きさの見積もりを与えよ．また，*member*, *find*, *merge* の操作に平均的にどのくらいの手間がかかるのか説明せよ．一般的にハッシュによる集合の表現はどのようなときに使うべきであるか説明せよ．

(4) *merge* の操作を $O(1)$ の手間で行うことができ，*find* の操作を平均 $O(\log n)$ の手間で行うことのできるデータ構造がある．どのようなデータ構造であり，*merge*, *find* の計算をどう行えばよいか，説明せよ．

Problem 2

Let A be a set of n elements, and $\Sigma = \{S_i\}_{i=1}^{m}$ be a set of subsets of A that satisfies

$$\begin{aligned} S_i \cap S_j &= \emptyset \qquad (i \neq j), \\ \bigcup_{i=1}^{m} S_i &= A. \end{aligned}$$

Consider the following operations on Σ.

- *member*(S, x): returns whether $S \in \Sigma$ contains $x \in A$ or not.
- *find*(x): returns the subset $S \in \Sigma$ to which $x \in A$ belongs.
- *merge*(S_i, S_j): removes $S_i \in \Sigma$ and $S_j \in \Sigma$ from Σ and adds $S_i \cup S_j$ to Σ.

Answer the following questions. Let M be the maximum cardinality of Σ.

(1) Consider representing a subset with a bit vector. Estimate necessary memory space to represent the subsets. Estimate average computational cost to perform each of *member*, *find*, and *merge* operations. In general in what cases bit vector representation should be used?

(2) Consider representing a subset with a linked list. Estimate necessary memory space to represent the subsets. Estimate average computational cost to perform each of *member*, *find*, and *merge* operations. In general in what cases linked list representation should be used?

(3) Consider representing the subsets by a hash table with elements as keys and subsets as values. Estimate necessary memory space to represent the subsets. Estimate average computational cost to perform each of *member*, *find*, and *merge* operations. In general in what cases hash table representation should be used?

(4) There is a data structure that can perform *merge* operation in $O(1)$ time and *find* operation in $O(\log n)$ on average. Explain the data structure and how to perform *merge* and *find* operations.

問題 3

決定性有限オートマトンを $M = (Q, \Sigma, \delta, q_0, F)$ と書く．ここで，Q は状態の有限集合，Σ は有限の入力アルファベット，$\delta : Q \times \Sigma \to Q$ は状態遷移関数，q_0 は初期状態，F は受理状態の集合とする．M によって受理される言語を $L(M)$ と書く．このとき，次の問いに答えよ．

(1) $\Sigma^* - L(M) = \{w \in \Sigma^* \mid w \notin L(M)\}$ は正則言語であることを証明せよ．

(2) 決定性有限オートマトン $M = (Q, \Sigma, \delta, q_0, F)$ が入力として与えられるとき，$L(M) = \emptyset$ となっているかどうかを判定するアルゴリズムを与え，その正当性について議論せよ．

(3) 2 つの決定性有限オートマトン $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ と $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ が入力として与えられるとき，$L(M_1) \subseteq L(M_2)$ となっているかを判定するアルゴリズムは存在するか．存在するとするならば，そのアルゴリズムの概要を示せ．存在しないとするならば，そのことを証明せよ．

Problem 3

Let a deterministic finite automaton $M = (Q, \Sigma, \delta, q_0, F)$. Here, Q is a finite set of states, Σ is a finite input alphabet, $\delta : Q \times \Sigma \to Q$ is a state transition function, q_0 is an initial state, and F is a set of accepting states. Write $L(M)$ for the language accepted by M. Answer the following questions.

(1) Prove that $\Sigma^* - L(M) = \{w \in \Sigma^* \mid w \notin L(M)\}$ is regular.

(2) Given a deterministic finite automaton $M = (Q, \Sigma, \delta, q_0, F)$ as an input, show an algorithm for testing $L(M) = \emptyset$ and discuss its correctness.

(3) Given two deterministic finite automata $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ as inputs, is there an algorithm for testing $L(M_1) \subseteq L(M_2)$? If there is, show briefly such an algorithm. If not, prove the non-existence.

問題 4

簡易ネットワークファイルサーバのCプログラムの一部を以下に示す．なお，本サーバプログラムは，クライアントと同一のアクセス権限下で常に動作すると仮定する．

```
/* the external declarations are omitted. */
typedef struct header {
  int  cmd_or_stat;
  int  sz;
} header;
typedef struct packet {
  header  hd;
  char    data[512];
} packet;
void mainbody() {
 int     sock;
 packet  pkt;
 sock = getConnection();
 while (rcv(sock, sizeof(header), &pkt)) {
   op(sock, &pkt);
   snd(sock, sizeof(header) + pkt.hd.sz,
       &pkt);
 }
 closeConnection(sock);
}
void op(int sock, packet *pkt) {
 int    fd;
 int    n = -1;
```

```
 switch (pkt->hd.cmd_or_stat) {
 case OP_OPEN:
  rcv(sock, pkt->hd.sz, pkt->data);
  pkt->data[pkt->hd.sz] = 0;
  fd = open(pkt->data, O_RDWR);
  if (fd > 0)  pkt->hd.cmd_or_stat = SUCCESS;
  else  pkt->hd.cmd_or_stat = FAIL;
  pkt->hd.sz = 0;
  break;
 case OP_READ:
  if (fd > 0) n = read(fd, pkt->data, 512);
  if (n < 0) {
    pkt->hd.cmd_or_stat = FAIL;
    pkt->hd.sz = 0;
  } else {
    pkt->hd.cmd_or_stat = SUCCESS;
    pkt->hd.sz = n;
  }
  break;
 case OP_CLOSE:
  /* the following code is omitted. */
 }
}
```

ここで，`getConnection`関数は，クライアントからのTCPコネクションを確立し，そのコネクションに関連づけられたファイルディスクリプタを返す．`closeConnection`関数は，引数で与えられたファイルディスクリプタに関連づけられているコネクションを閉じる．`snd`および`rcv`関数は，送信，受信関数であり，第一引数にコネクションに関連づけられたファイルディスクリプタ，第二引数に送信あるいは受信バイト数，第三引数にデータのアドレスを与える．これら4つの関数実行は常に成功すると仮定する．

`open`システム関数は，第一引数にファイル名をとる．正しくファイルが開けばファイルディスクリプタを返し，そうでなければ`-1`を返す．`read`システム関数は，第一引数にデータを読み込むファイルディスクリプタ，第二引数にデータを格納するメモリアドレス，第三引数に読み込むデータサイズをとる．返値は，正しく読み込めれば読み込んだサイズ，エラーが生じれば`-1`が返る．以下の問いに答えよ．

(1) `op`関数内に，変数宣言に致命的な問題が一つ，セキュリティホールが一つ存在する．これら2つの問題点およびセキュリティホールのメカニズムを説明し，プログラムを直せ．

(2) クライアントプログラムとして，ネットワークファイルサーバ側の`/home/foo/data.txt`ファイルを読み出して，クライアント側で表示するプログラムを書け．なお，サーバとのコネクションを確立するための関数`reqConnection`が提供されていると仮定する．本関数は，引数を持たず，コネクションに関連づけられたファイルディスクリプタを返す．本関数実行は常に成功すると仮定する．

(3) `OP_READ`コマンドは，連続して要求されることが多いと仮定して，応答性能を上げるためにプログラムをどのように変更すれば良いか示せ．

Problem 4

A part of a simple network file server program written in C is shown below. Assume that the server program always runs under the same access rights of the client.

```
/* the external declarations are omitted. */
typedef struct header {
  int  cmd_or_stat;
  int  sz;
} header;
typedef struct packet {
  header  hd;
  char    data[512];
} packet;
void mainbody() {
 int     sock;
 packet  pkt;
 sock = getConnection();
 while (rcv(sock, sizeof(header), &pkt)) {
   op(sock, &pkt);
   snd(sock, sizeof(header) + pkt.hd.sz,
       &pkt);
 }
 closeConnection(sock);
}
void op(int sock, packet *pkt) {
 int    fd;
 int    n = -1;
```

```
 switch (pkt->hd.cmd_or_stat) {
 case OP_OPEN:
  rcv(sock, pkt->hd.sz, pkt->data);
  pkt->data[pkt->hd.sz] = 0;
  fd = open(pkt->data, O_RDWR);
  if (fd > 0)  pkt->hd.cmd_or_stat = SUCCESS;
  else  pkt->hd.cmd_or_stat = FAIL;
  pkt->hd.sz = 0;
  break;
 case OP_READ:
  if (fd > 0) n = read(fd, pkt->data, 512);
  if (n < 0) {
    pkt->hd.cmd_or_stat = FAIL;
    pkt->hd.sz = 0;
  } else {
    pkt->hd.cmd_or_stat = SUCCESS;
    pkt->hd.sz = n;
  }
  break;
 case OP_CLOSE:
  /* the following code is omitted. */
 }
}
```

The `getConnection` function establishes the TCP connection requested by the client and returns the file descriptor associated with the connection. The `closeConnection` function closes the connection associated with the file descriptor given by its argument. The `snd` and `rcv` functions are for sending and receiving, respectively. In those functions, the first, second, and third arguments take the file descriptor associated with a connection, size in byte, and data address, respectively. Assume that the above four functions are always executed without errors.

The `open` system function takes the file name given by the first argument. It returns the file descriptor of the file on success, and `-1` on failure. The `read` system function reads the data using the file descriptor specified by the first argument, and stores it to the address specified by the second argument. The data size to be read is specified by the third argument. It returns the size of read data on success, and `-1` on failure. Answer the following questions.

(1) There is a fatal error of variable declaration and a security hole in the `op` function. Explain those problems and the security hole mechanism. Fix the program.

(2) Write a client program in which the file `/home/foo/data.txt` in the network file server is read and displayed at the client. The program may use the `reqConnection` function which establishes the TCP connection between the client and the server. The function takes no argument and returns the file descriptor of the connection. Assume that the execution of this function never fails.

(3) Assume that the `OP_READ` command may be requested continuously. Explain how the program should be modified in order to improve the response time of the command.

平成21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 II

平成20年8月26日

13:30 – 16:00

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 the answer sheets and the problem booklet out of the examination room.

下欄に受験番号を記入すること.

Fill the following blank with your examinee's number.

受験番号	No.

問題 1

A は n 次対称正定値行列とする．ただし，n 次正方行列 A が正定値とは，任意の n 次列ベクトル $\boldsymbol{x} \neq \boldsymbol{0}_n$ に対して $\boldsymbol{x}^\top A\boldsymbol{x} > 0$ となることをいう．なお，$\boldsymbol{x}^\top$ は $\boldsymbol{x}$ の転置，$\boldsymbol{0}_n$ はすべての要素が 0 であるような n 次列ベクトルを表す．また，I_n は n 次単位行列を表す．

以下の問いに答えよ．

(1) 行列 A を

$$A = \begin{pmatrix} \alpha_{11} & \boldsymbol{a}_1^\top \\ \boldsymbol{a}_1 & A_1 \end{pmatrix}$$

とする．ただし α_{11} はスカラー，$\boldsymbol{a}_1$ は $n-1$ 次列ベクトル，A_1 は $n-1$ 次の正方行列である．A の正定値性から $\alpha_{11} > 0$ となることを示せ．

(2) n 次正方行列 L_1, D_1 を

$$L_1 = \begin{pmatrix} \lambda_{11} & \boldsymbol{0}_{n-1}^\top \\ \boldsymbol{l}_1 & I_{n-1} \end{pmatrix}, \qquad D_1 = \begin{pmatrix} 1 & \boldsymbol{0}_{n-1}^\top \\ \boldsymbol{0}_{n-1} & A_1' \end{pmatrix}$$

とする．ただし λ_{11} はスカラー，$\boldsymbol{l}_1$ は $n-1$ 次列ベクトル，A_1' は $n-1$ 次の正方行列である．$A = L_1 D_1 L_1^\top$ となるように $\lambda_{11}, \boldsymbol{l}_1, A_1'$ を定め，問い (1) の $\alpha_{11}, \boldsymbol{a}_1, A_1$ を用いて表せ．ただし $\lambda_{11} > 0$ となるようにせよ．

(3) A_1' も対称正定値行列であることを示せ．

(4) 以上の結果を用いて，$A = LL^\top$ となるような下三角行列 L を計算するアルゴリズムを述べよ．

Problem 1

Let A be an $n \times n$ symmetric positive definite matrix. An $n \times n$ matrix A is said to be positive definite when $\boldsymbol{x}^\top A\boldsymbol{x} > 0$ holds for any column vector $\boldsymbol{x} \neq \mathbf{0}_n$ of length n. Note that $\boldsymbol{x}^\top$ represents the transpose of $\boldsymbol{x}$, and $\mathbf{0}_n$ represents a column vector of length n in which all elements are zeros. In the following questions, I_n represents an $n \times n$ identity matrix.

Answer the following questions.

(1) Let A be represented as

$$A = \begin{pmatrix} \alpha_{11} & \boldsymbol{a}_1^\top \\ \boldsymbol{a}_1 & A_1 \end{pmatrix},$$

where α_{11} is a scalar, $\boldsymbol{a}_1$ is a column vector of length $n-1$, and A_1 is an $(n-1) \times (n-1)$ matrix. Show $\alpha_{11} > 0$ from positive definiteness of A.

(2) Let L_1 and D_1 be $n \times n$ matrices defined as

$$L_1 = \begin{pmatrix} \lambda_{11} & \mathbf{0}_{n-1}^\top \\ \boldsymbol{l}_1 & I_{n-1} \end{pmatrix}, \qquad D_1 = \begin{pmatrix} 1 & \mathbf{0}_{n-1}^\top \\ \mathbf{0}_{n-1} & A_1' \end{pmatrix},$$

where λ_{11} is a scalar, $\boldsymbol{l}_1$ is a column vector of length $n-1$, and A_1' is an $(n-1) \times (n-1)$ matrix. Determine λ_{11}, $\boldsymbol{l}_1$, and A_1' so that $A = L_1 D_1 L_1^\top$ and $\lambda_{11} > 0$, and represent λ_{11}, $\boldsymbol{l}_1$, and A_1' in terms of α_{11}, $\boldsymbol{a}_1$ and A_1 in question (1).

(3) Show that A_1' is also a symmetric positive definite matrix.

(4) Give an algorithm to compute a lower triangular matrix L that satisfies $A = LL^\top$, using the above results.

問題 2

n 本からなる 2 次元ベクトルの列 $X = \langle \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n \rangle$ が与えられている．ここで，X の i 番目から j 番目のベクトルからなる部分列（ただし $i \le j$ とする）を $X[i..j] = \langle \boldsymbol{v}_i, \boldsymbol{v}_{i+1}, \ldots, \boldsymbol{v}_j \rangle$ と表すものとする．このとき，X に対し，$\boldsymbol{g}(i,j)$ $(i \le j)$ を，部分列 $X[i..j]$ の平均，すなわち

$$\boldsymbol{g}(i,j) = \frac{1}{j-i+1} \sum_{l=i}^{j} \boldsymbol{v}_l$$

と定義する．

X の k 個の部分列への分割を k 分割といい，X の $X[1..a_1], X[a_1+1..a_2], \ldots, X[a_{k-1}+1..n]$ への k 分割を $\mathcal{D}(a_1, a_2, \ldots, a_{k-1})$ で表すものとする（ただし $0 < a_1 < a_2 < \cdots < a_{k-1} < n$ とする）．また，$\mathcal{D} = \mathcal{D}(a_1, a_2, \ldots, a_{k-1})$ によってできる各部分列の平均のノルムを求め，それらの平均をとったもの

$$\frac{1}{k} \sum_{i=1}^{k} \|\boldsymbol{g}(a_{i-1}+1, a_i)\|$$

を $H(X,\mathcal{D})$ と表すものとする．ただし，ここでは $a_0 = 0$，$a_k = n$ としている．このとき，$H(X,\mathcal{D})$ の値が最小となるような k 分割 $\mathcal{D}_{min}(X,k)$ を求めたい．また，そのときの最小値を $S_k(X)$ と表すものとする．

このとき，以下の問いに答えよ．

(1) 列 X を

$$X = \left\langle \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 4 \\ 7 \end{pmatrix}, \begin{pmatrix} 7 \\ 20 \end{pmatrix}, \begin{pmatrix} 3 \\ 4 \end{pmatrix} \right\rangle$$

としたとき，$H(X,\mathcal{D}(2))$ を求めよ．

(2) X に対する $O(n)$ 時間の前処理を行うことによって，任意の i，j $(1 \le i \le j \le n)$ の組に対し $\boldsymbol{g}(i,j)$ を定数時間で求めることができることを示せ．

(3) $k-1 \le i < n$ をみたすすべての i に対し，$S_{k-1}(X[1..i])$ の値がわかっている場合に，$S_k(X)$ を $O(n)$ 時間で求めるアルゴリズムを示せ．

(4) $\mathcal{D}_{min}(X,k)$ を効率よく求めるアルゴリズムとその計算量を示せ．

Problem 2

Let $X = \langle \boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_n \rangle$ be a sequence of n vectors of length 2. The part of the sequence X from the i-th vector to the j-th vector ($i \le j$), *i.e.*, $\langle \boldsymbol{v}_i, \boldsymbol{v}_{i+1}, \dots, \boldsymbol{v}_j \rangle$, is called a subsequence of X and denoted by $X[i..j]$. We define $\boldsymbol{g}(i,j)$ ($i \le j$) as an average of subsequence $X[i..j]$, *i.e.*,

$$\boldsymbol{g}(i,j) = \frac{1}{j-i+1} \sum_{l=i}^{j} \boldsymbol{v}_l.$$

A division of X into k subsequences is called a k-division. Let $\mathcal{D}(a_1, a_2, \dots, a_{k-1})$ denote the k-division that divides X into $X[1..a_1], X[a_1+1..a_2], \dots, X[a_{k-1}+1..n]$ (where $0 < a_1 < a_2 < \cdots < a_{k-1} < n$ is assumed). Let $H(X,\mathcal{D})$ denote an average of the norms of the averages of the subsequences obtained by the k-division $\mathcal{D} = \mathcal{D}(a_1, a_2, \dots, a_{k-1})$, *i.e.*,

$$\frac{1}{k} \sum_{i=1}^{k} \| \boldsymbol{g}(a_{i-1}+1, a_i) \|$$

where $a_0 = 0$ and $a_k = n$. We want to find the k-division $\mathcal{D}_{min}(X,k)$ with the minimum $H(X,\mathcal{D})$ value. Let $S_k(X)$ denote the minimum value.

Answer the following questions.

(1) Compute $H(X,\mathcal{D}(2))$ in case

$$X = \left\langle \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 4 \\ 7 \end{pmatrix}, \begin{pmatrix} 7 \\ 20 \end{pmatrix}, \begin{pmatrix} 3 \\ 4 \end{pmatrix} \right\rangle.$$

(2) Show that $\boldsymbol{g}(i,j)$ can be computed in constant time for any i and j ($1 \le i \le j \le n$), after an $O(n)$-time preprocessing on X.

(3) In case we are given the values of $S_{k-1}(X[1..i])$ for all i such that $k-1 \le i < n$, describe an algorithm that computes $S_k(X)$ in $O(n)$ time.

(4) Describe an algorithm that computes $\mathcal{D}_{min}(X,k)$ efficiently, and show the computational time complexity of it.

問題 3

プログラミング言語設計に関して以下の問いに答えよ．

(1) 「スタックモジュール」を作ることを考える．データ構造の定義と，以下の操作の定義を書き下せ．(言語は，C 言語を用いよ.)

create 空のスタックを作る

push スタックにデータを 1 つ挿入

pop スタックトップのデータを返すとともに，そのデータをスタックから削除

(2) 問い (1) で定義したようなデータ構造を，モジュールの外から直接変更することを考える．(たとえば，スタックトップから 2 番目以降にあるデータを，上の操作を使わずに，新たなデータに置き換える.)

(a) このような変更が許されることの問題点は何か？

(b) このような変更を禁止するためには，プログラミング言語はどのような機能を提供すればよいか？

Problem 3

Answer the following questions relevant to programming language design.

(1) Consider implementing a "stack module." Write down the definition of a data structure for it and the following operations on the data structure. (Use the C language.)

create creates an empty stack

push inserts a data item to the stack

pop returns the data at the top of the stack and removes it from the stack

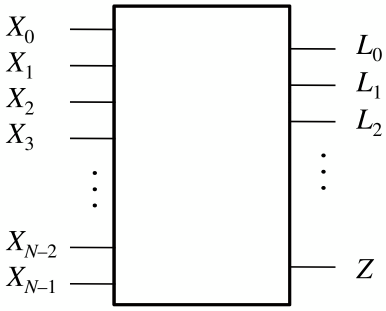
(2) Consider directly modifying the data structure defined in question (1) *from the outside of the module.* (For example, consider replacing, with a new data item, a data item located at the second or deeper place from the top of the stack without using the above operations.)

(a) What is a problem of permitting such modification?

(b) In order to forbid this, what feature should a programming language provide?

問題 4

N ビットのデータに対して，MSB (Most Significant Bit) から最初に 1 が立っているビットまでのビット距離を出力する論理回路を，以下の手順で設計せよ．なお，上記論理回路への入力は N ビットのデータ X であり，出力はビット距離を 2 進法で表す L と，X のすべてのビットが 0 であることを示す Z である．ただし，X のすべてのビットが 0 のとき，L の値は任意とする．以下に $N = 20$ の例を示す．



		MSB … LSB
Example 1	X	0 0 0 0 1 0 1 1 0 0 1 1 0 0 1 0 1 1 1 1
	L	0 0 1 0 0
	Z	0
Example 2	X	1 0 0 1 1 0 0 0 0 1 0 0 1 0 0 0 1 1 0 0
	L	0 0 0 0 0
	Z	0
Example 3	X	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
	L	* * * * *
	Z	1

* : don't care

以下の問いに答えよ．

(1) $N = 4$ の場合の論理回路の真理値表を示せ．

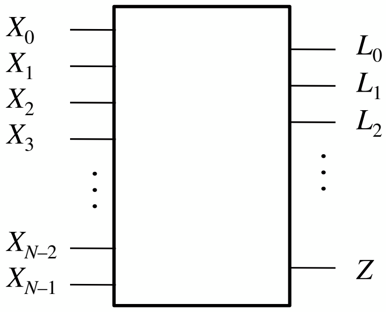
(2) $N = 4$ の場合の論理回路を，AND, OR, NOT ゲートを用いて構成せよ．

(3) $N = 16$ の場合の論理回路を，問い (2) で求めた回路と AND, OR, NOT ゲートを組み合わせて構成せよ．

(4) 任意の N に対し，遅延時間が $O(\log N)$ である論理回路の構成法を示せ．

Problem 4

Design by the following steps a logic circuit that calculates the bit-distance from MSB (Most Significant Bit) to the first bit position where the bit is 1. Let X be the input data of N bits to the above mentioned logic circuit, L be the bit-distance in the binary number system, and Z indicate the situation where all the bits in X are 0. When all the bits in X are 0, the value of L is arbitrary. The following figure shows a few examples for $N = 20$.



		MSB … LSB
Example 1	X	0 0 0 0 1 0 1 1 0 0 1 1 0 0 1 0 1 1 1 1
	L	0 0 1 0 0
	Z	0
Example 2	X	1 0 0 1 1 0 0 0 0 1 0 0 1 0 0 0 1 1 0 0
	L	0 0 0 0 0
	Z	0
Example 3	X	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
	L	* * * * *
	Z	1

* : don’t care

Answer the following questions.

(1) Show the truth table of the case $N = 4$.

(2) Design a logic circuit of the case $N = 4$ using AND, OR and NOT gates.

(3) Design a logic circuit of the case $N = 16$ using some copies of the circuit designed in question (2) and AND, OR and NOT gates.

(4) Show a construction method for any N where the latency of the logic circuit is $O(\log N)$.