平 成 ２８ 年 度

名古屋大学大学院情報科学研究科
計算機数理科学専攻
入 学 試 験 問 題

専　　門

平成２７年８月６日（木）
１２：３０～１５：３０

注 意 事 項

１．試験開始の合図があるまでは、この問題冊子を開いてはならない。

２．試験終了まで退出できない。

３．外国人留学生は、日本語から母語への辞書１冊に限り使用してよい。
電子辞書の持ち込みは認めない。

４．問題冊子、解答用紙３枚、草稿用紙３枚が配布されていることを確認せよ。

５．問題は、線形代数、微分積分、離散数学、グラフ理論、数理論理学、確率・統計、量子力学、アルゴリズム設計法、オートマトン理論、プログラミングの１０題からなる。このうち**３題を選択して**解答せよ。

選択した問題名または問題番号を解答用紙の指定欄に記入せよ。

６．解答用紙の指定欄に受験番号を必ず記入せよ。解答用紙に受験者の氏名を記入してはならない。

７．解答用紙は試験終了後に３枚とも提出せよ。

８．問題冊子、草稿用紙は試験終了後に持ち帰ってよい。

問題1. (線形代数)

a を実数 (real number) とし,

$$A = \begin{pmatrix} 3-2a & a-2 & a & a \\ 2-2a & a-1 & a & a \\ a & -a & 1-a & 0 \\ 2-2a & 2a-2 & a & 1 \end{pmatrix}$$

とする. A が対角化可能 (diagonalizable) となるための必要十分条件を求め, 対角化せよ.

問題 **2.** (微分積分)

以下の各問に答えよ.

(1) 関数 $f(x)$ を $f(x) = \dfrac{1}{x} + \dfrac{1}{(x-1)^2} + \dfrac{1}{x-2}$ とする. 以下の小問に答えよ.

(i) $x < 0$ に対して $f(x) < 0$ となることを証明せよ.

(ii) 方程式 $f(x) = 0$ の実数解の個数 (number of real solutions) を求めよ.

(2) 非負整数 n に対して $I_n = \displaystyle\int_0^{\pi/2} (\sin x)^n dx$ と定義する. 以下の小問に答えよ.

(i) $I_n = \displaystyle\int_0^{\pi/2} (\cos x)^n dx$ が成立することを証明せよ.

(ii) n が 2 以上ならば $I_n = \dfrac{n-1}{n} I_{n-2}$ が成立することを証明せよ.

(iii) $\displaystyle\int_0^{\pi/2} (\sin x)^5 dx$, $\displaystyle\int_0^{\pi/2} (\cos x)^6 dx$ の値をそれぞれ求めよ.

問題 3. (離散数学)

以下の各問に答えよ.

(1) p を素数とする.

(i) $p \geq 3$ のとき,

$$\sum_{a=1}^{p-1} a \equiv 0 \mod p$$

を証明せよ.

(ii) $p \geq 5$ のとき,

$$\sum_{1 \leq a < b \leq p-1} ab \equiv 0 \mod p$$

を証明せよ.

(iii) x に関する多項式:

$$f(x) = \prod_{a=1}^{p-1} (x - a)$$

の係数は, 定数項と最高次を除いて p の倍数であることを証明せよ.

(2) 自然数 a, b に対して, それらの最大公約数を (a, b), 最小公倍数を $\{a, b\}$ で表す.

$$\{(a, b), (a, c)\} = (a, \{b, c\}), \quad (\{a, b\}, \{a, c\}) = \{a, (b, c)\}$$

を証明せよ.

問題 4. (グラフ理論)

以下の各問に答えよ．なお，グラフは有限であるとする．

(1) n (≥ 2) 個のフライトと各フライトに対する出発空港，到着空港，出発時刻，到着時刻が与えられる．フライト i の到着空港とフライト j の出発空港が同じで，しかも j の出発時刻が i の到着時刻の 30 分後以降であるときに i から j に乗り継ぎ可能であるとする．各フライト i を頂点とみなし，フライト i からフライト j に乗り継ぎ可能であるときかつそのときに限り i から j への有向辺 (i,j) が存在する有向グラフを考える．このようなグラフに有向閉路があるか否かを理由とともに述べよ．

(2) n 頂点からなる有向グラフ $G=(V,A)$ に有向閉路がないとき，各頂点 $v \in V$ にラベル $l(v)$ ($l: V \to \{1,2,\ldots,n\}$) をつけて

$$(u,v) \in A \implies l(u) < l(v)$$
および
$$u \neq v \implies l(u) \neq l(v)$$

を満たすようにできることを，数学的帰納法を用いて示せ．

(3) 各辺 $(u,v) \in E$ に長さ $d(u,v)$ (>0) が与えられた完全無向グラフ $G=(V,E)$ を考える．V の各頂点をちょうど一度ずつ訪れる閉路を巡回路とよび，巡回路上の辺の長さの総和（巡回路長）を最小にする巡回路を求める問題を巡回セールスマン問題 (traveling salesman problem, TSP) とよぶ．また，$T \subseteq E$ に対してグラフ (V,T) の連結成分が 1 つで閉路を持たないとき，T を全域木とよび，含まれる辺の長さの総和が最小の全域木を最小木 (minimum spanning tree, MST) と呼ぶ．最小の巡回路長 $f^*_{\mathrm{TSP}}(G,d)$ と最小木の辺の長さの総和 $f^*_{\mathrm{MST}}(G,d)$ の間に成立する大小関係を理由とともに述べよ．

用語．フライト: flight, 出発空港: departure airport, 到着空港: arrival airport, 出発時刻: departure time, 到着時刻: arrival time, 乗り継ぎ: transit, グラフ: graph, 有向: directed, 頂点: vertex, 辺: edge, 閉路: cycle, 完全無向グラフ: complete undirected graph, 巡回路: tour, 連結成分: connected component, 全域木: spanning tree.

問題5. (数理論理学)

Aを非可算個 (uncountable) の実数の集合とする．Aの集積点 (accumulation points) の集合は非可算になるか？

問題6. (確率・統計)

事象 A の確率を $\Pr(A)$, 確率変数 Z の期待値を $\mathbb{E}[Z]$ と表す. 以下の各問に答えよ. ただし, 非負実数に値をとる確率変数 (non-negative real valued random variable) Z と正の実数 c に対して成り立つマルコフの不等式

$$\Pr(Z \geq c) \leq \frac{\mathbb{E}[Z]}{c}$$

を証明なしに用いてよい.

(1) a を実数, t を正の実数とする. 任意の確率変数 Y に対して

$$\Pr(Y \geq a) \leq e^{-ta}\mathbb{E}[e^{tY}]$$

が成り立つことを示せ.

(2) 確率変数 X は確率 p で 1 をとり, 確率 $1-p$ で 0 をとるとする. ここで $0 < p < 1$ とする. 正の実数 t に対して

$$\mathbb{E}[e^{tX}] \leq e^{p(e^t-1)}$$

が成り立つことを示せ.

(3) $X_1, \ldots, X_n$ は X と同一の分布に独立にしたがう確率変数とする. δ を正の実数とし, $h(\delta) = (1+\delta)\log(1+\delta) - \delta$ とおく. このとき

$$\Pr\left(\frac{1}{n}\sum_{i=1}^{n} X_i \geq (1+\delta)\,p\right) \leq e^{-nph(\delta)}$$

が成り立つことを示せ.

問題 7. (量子力学)

$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$ (ただし, $i = \sqrt{-1}$) とするとき, 以下の各問に答えよ.

(1) t を実数とする.

(i) 行列 $M(t) = e^{-itX}$ を成分で記述せよ.

(ii) $M(t)$ が観測可能量 (observable) となるような t の値を答えよ.

(iii) $M(t)$ がユニタリ (unitary) となるような t の値を答えよ.

(2) Y の固有値 (eigenvalue) と対応する固有状態 (eigenstate) を求めよ.

(3) $U(\tau) = e^{-i\tau X}$ を時刻 $t = 0$ から時刻 $t = \tau \geq 0$ までの閉じた量子力学系の時間発展 (time evolution) を記述するユニタリ作用素とする.

(i) 時刻 $t = 0$ での系の状態が固有値 $+1$ に対応する Y の固有状態であるとき, 時刻 $t = \tau$ での系の状態を求めよ.

(ii) $Y = +1$ である確率 (probability) が $1/2$ であるような τ を求めよ.

問題8. (アルゴリズム設計法)

英文の文章の左右の端をそろえて見栄えよく印刷するための改行 (line break) 位置を決めたい．ここで，改行は単語 (word) と単語の間のみで行うものとし，最終行の右端はそろえなくてよいものとする．k 番目と $k+1$ 番目の単語の間を改行位置 k とよび，実際に改行を行う改行位置の系列 $k_1, k_2, \ldots, k_l$ で配置を定める．たとえば図1と図2は同じ文章の異なるレイアウトであり，図1の改行位置系列は 5, 7, 11, 図2の系列は 6, 9, 13 である．

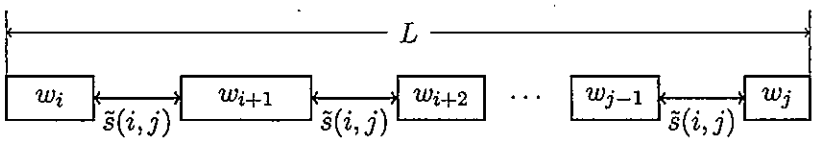
```
It is very important to
design sophisticated
algorithms that run efficiently
for large instances.
```

図1

```
It is very important to design
sophisticated algorithms that
run efficiently for large
instances.
```

図2

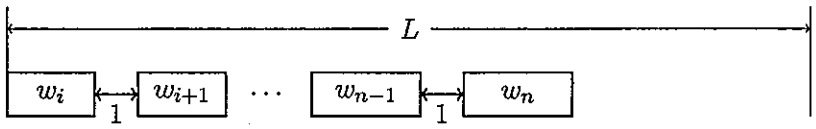
文章は n 個の単語からなり，i 番目の単語の幅 w_i $(i = 1, 2, \ldots, n)$ と行の幅 L が与えられる．ただし L は十分大きく，1行に2単語以上入る幅である．
i 番目から j 番目まで $(i < j)$ の単語を1行に配置したとき，次の図のように単語間に均等にスペースを配置するときのスペース幅を $\tilde{s}(i,j)$ とする．



また，$j = i$ のときは便宜上 $\tilde{s}(i,i) = +\infty$ とし，

$$\tilde{s}(i,j) = \begin{cases} \dfrac{L - (w_i + w_{i+1} + \cdots + w_j)}{j - i}, & \text{if } i < j \\ +\infty, & \text{if } i = j \end{cases}$$

と定義する．$j = n$（最終行）において $\tilde{s}(i,n)$ が1を超える場合には次の図のように単語間のスペースとして1を用いる．



以上のことを考慮して，$s(i,j)$ $(i \leq j)$ を

$$s(i,j) = \begin{cases} \tilde{s}(i,j), & \text{if } j < n \\ \min\{\tilde{s}(i,j), 1\}, & \text{if } j = n \end{cases}$$

と定義し，これを単語間のスペースに用いて配置する．
このとき，i 番目から j 番目まで $(i \leq j)$ の単語を1行に配置したときのコスト $c(i,j)$ を

$$c(i,j) = \begin{cases} +\infty, & \text{if } s(i,j) \leq 0 \\ |s(i,j) - 1|, & \text{otherwise} \end{cases}$$

と定める．さらに，改行位置系列 $k_1, k_2, \ldots, k_l$ で配置を定めるときのレイアウトのコスト z を

$$z = c(1, k_1) + c(k_1 + 1, k_2) + \cdots + c(k_l + 1, n)$$

と定め，その最小化を図る．

以下の各問に答えよ．

(1) 図1のレイアウトを考える．この図に現れる単語の幅を文字数で定義すると，以下の表が得られる．

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14
w_i	2	2	4	9	2	6	13	10	4	3	11	3	5	9

ここで，ピリオド "." を無視することで，14個めの単語 instances の幅を $w_{14} = 9$ としている．行幅を $L = 30$ とするとき，各行のコスト，および文章全体のコスト z を求めよ．

(2) すべての i, j の組 $(1 \leq i < j \leq n)$ に対して $c(i, j)$ の値を計算する効率の良い方法を述べ，その計算量を評価せよ．

(3) 改行位置系列を全列挙することで z 最小のレイアウトを求める方法の計算量を述べよ．

(4) i 番目から n 番目までの $n - i + 1$ 個の単語からなる文章をレイアウトするときの最小コストを $g(i)$ とする．$g(i)$ を計算する動的計画法 (dynamic programming) の漸化式 (recurrence formula) を書け．（必要であれば $g(n + 1) = 0$ としてよい．）

(5) $g(1), \ldots, g(n)$ を具体的に計算する方法を説明し，その計算量を評価せよ．

(6) n 個の単語からなる文章をレイアウトする際のコスト z の最小値，ならびに，それを実現する改行位置系列の求め方を説明せよ．

問題 9. (オートマトン理論)

アルファベット (alphabet) $\Sigma = \{a, b\}$ とする．以下の各問に答えよ．

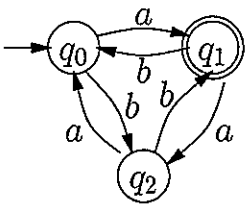
(1) 次の言語を認識する決定性有限オートマトン (deterministic finite automaton) を図示せよ．

(i) $L_1 = \{w \mid w \in \Sigma^*,\ w \text{ は } bb \text{ を含む}\}$

(ii) $L_2 = \{w \mid w \in \Sigma^*,\ w \text{ は } aba \text{ を含まない}\}$

(iii) $L_3 = L_1 \cap L_2$

(2) $\Gamma = \{0, 1, 2\}$ とする．$h(0) = a$, $h(1) = ba$, $h(2) = aa$ で定まる準同型写像 (homomorphism) $h : \Gamma^* \to \Sigma^*$，ならびに，次の図で示される有限オートマトン M について，言語 $h^{-1}(L(M))$ を認識する有限オートマトンを図示せよ．



(3) 言語上の連接 $R.S$ ならびにべき乗 R^n $(n \geq 0)$ を，以下のように定義する．

$$R.S = \{rs \mid r \in R,\ s \in S\}$$
$$R^0 = \{\varepsilon\},\quad R^{i+1} = R^i.R\ (i = 0, 1, 2, \cdots)$$

このとき，任意の $n \geq 0$ について $R^{n+1} = R.R^n$ であることを数学的帰納法 (mathematical induction) を用いて示せ．ここで，以下の性質は既に証明されているとして用いても構わない．

$$\{\varepsilon\}.R = R.\{\varepsilon\} = R$$
$$(R.S).T = R.(S.T)$$

問題 10. (プログラミング)

2分木は各頂点が高々2つの頂点を子に持つ根付き木である．2分木の各頂点はデータを保持し，その子は左右で区別される．以下では各頂点がデータとして整数を保持する2分木を考える．図1に2分木の例を示す．

図 1: 2分木の例

C言語プログラムでは以下に定義される構造体により2分木を表現することができる．

```
struct vertex {
  int value;
  struct vertex *left;
  struct vertex *right;
};
```

この構造体は1つの頂点を表現し，その頂点が保持する整数を格納するメンバ value，左の子へのポインタを格納するメンバ left，右の子へのポインタを格納するメンバ right から構成される．この構造体で表現された頂点の左の子，右の子が存在しない場合はメンバ left，right にそれぞれ値 NULL を代入する．

ソースコード1は上述の構造体を利用して表現された2分木を取り扱うC言語プログラムである．大域変数 tree は2分木を参照するポインタ変数であり，宣言時の初期値を NULL とする．すなわち，頂点を持たない空の2分木を NULL で表す．insert 関数，eliminate 関数はポインタ tree が指す2分木に対して引数で与えられた整数を保持する頂点をそれぞれ挿入，削除する関数である．create 関数は引数で与えられた整数を保持する頂点を作成する関数である．display 関数は引数で与えられたポインタが指す2分木（もしくは2分部分木）に対して頂点が保持する整数を先行順で出力する関数であり，再帰的に定義されている．

ソースコード1について以下の問いに答えよ．

[1] main 関数を 18 行目の末尾まで実行した時点，ならびに 19 行目の末尾まで実行した時点におけるポインタ tree が指す2分木を図1の記法に従いそれぞれ図示せよ．

[2] 21 行目の末尾まで実行した時点において tree が指す2分木が図1 (b) に示される2分木の根を参照するように 21 行目にデータの挿入・削除の操作を記述したい．21 行目の [(ア)] に書くべき命令の列を1つ示せ．ただし，命令の列は以下の条件を満たすものとする．

- 命令は insert(x); または eliminate(y); のみである．ただし，x および y は整数である．
- insert 関数や eliminate 関数の引数に与えられる整数には同じものは高々1回しか現れない．

[3] 三項演算子 ? : は「式1?式2:式3」のように記述されて式を構成する.「式1」を評価した結果が0でないときは「式2」を,0であるときは「式3」を評価した結果を式の値とする.例えば,式 x <= 0 ? x+1 : x+2 を評価した値は x の値が −1 のときには 0 となり,5 のときは 7 となる.この三項演算子を用いて,53 行目から 56 行目のコード

```
if( x < p->value )
  p = p->left;
else
  p = p->right;
```

と同様の動作をする代入文を以下の［　(イ)　］を埋めて完成させよ.

```
p = [        (イ)        ];
```

[4] ソースコード 1 の eliminate 関数の定義にはメモリへの不正アクセスを発生させる可能性がある.例えば,19 行目と次の行の間で eliminate(10); を実行した場合に不正アクセスが発生する.eliminate 関数の定義の 1 箇所に命令を追加してその問題を解決せよ.なお,解答は「○○行目と次の行の間に△△を挿入」という形式で記すこと.

[5] ソースコード 1 の eliminate 関数では不要になったメモリを解放していない.このままで挿入・削除の実行を繰り返した場合にどのような問題が起こりうるかを説明せよ.さらに,不要になったメモリを解放するように,eliminate 関数の定義の中の 2 箇所にそれぞれ命令を追加せよ.解答は「○○行目と次の行の間に△△を挿入」という形式で記せ.また,メモリの解放には標準ライブラリ関数である free 関数を使用すること.

[6] 21 行目の末尾まで実行した時点でポインタ tree が指す 2 分木は図 1 (b) になっている.このあと,22 行目を実行した際に出力される文字列を示せ.

[7] display 関数を実行した際に数が大きい順に出力されるようその定義を変更したい.以下の関数定義の中の［　(ウ)　］～［　(オ)　］を埋めて変更せよ.頂点に保持される整数を出力する場合には printf("%d,", p->value) を記述すること.

```
void display(struct vertex *p){
  if( p == NULL ) return;
  [        (ウ)        ];
  [        (エ)        ];
  [        (オ)        ];
  return;
}
```

[8] 2 分木に指定した数が存在しているかどうかを探索する member 関数を以下の要件を満たすように作成したい.

- 引数として整数 x を受け取る.
- ポインタ tree が指す 2 分木に x を保持する頂点が存在する場合には 1 を,そうでない場合には 0 を返す.

これらの条件を満たすように以下の関数定義の中の［　（カ）　］～［　（ケ）　］を埋めてmember関数の定義を完成させよ．

```
int member(int x){
  struct vertex *p;
  p = tree;
  while( p != NULL ){
    if( [　　(カ)　　] ) return 1;
    if( [　　(キ)　　] )
      [　　(ク)　　];
    else
      [　　(ケ)　　];
  }
  return 0;
}
```

Translation of technical terms

2分木	binary tree	関数	function
頂点	vertex	部分木	subtree
子	child	先行順	preorder
根付き木	rooted tree	出力する	print out
C言語	C programming language	再帰的に	recursively
構造体	structure	操作	operation
メンバ	member	命令	statement
ソースコード	source code	列	sequence
プログラム	program	三項演算子	ternary operator
大域変数	global variable	式	expression
格納する	store	代入文	assignment statement
参照する	refer	メモリ	memory
ポインタ	pointer	不正アクセス	illegal access
変数	variable	発生させる	raise
宣言	declaration	実行する	execute
初期値	initial value	解放する	free
空	empty	標準ライブラリ関数	standard library function
引数	argument	文字列	string
挿入する	insert	関数定義	function definition
削除する	delete	探索する	search

ソースコード 1: 2 分木を処理する C 言語プログラム

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct vertex {
5    int value;
6    struct vertex *left;
7    struct vertex *right;
8  };
9
10 struct vertex *tree = NULL;
11
12 struct vertex *create(int x);
13 void insert(int x);
14 void eliminate(int x);
15 void display(struct vertex *p);
16
17 int main(){
18   insert(3); insert(12); insert(18); insert(11); insert(14);
19   eliminate(12); insert(2); insert(3);
20   insert(6); eliminate(3); insert(5);
21   [          (ア)          ]
22   display(tree);
23   return 0;
24 }
25
26 struct vertex *create(int x){
27   struct vertex *p;
28   p = (struct vertex *) malloc(sizeof(struct vertex));
29   p->value = x;
30   p->left = NULL;
31   p->right = NULL;
32   return p;
33 }
34
35 void insert(int x){
36   struct vertex *p;
37
38   if( tree == NULL ){
39     tree = create(x);
40     return;
41   }
42
43   p = tree;
44   do{
45     if( p->value == x ) break;
46     else if( x < p->value && p->left == NULL ){
47       p->left = create(x);
48       break;
49     }else if( p->value < x && p->right == NULL ){
50       p->right = create(x);
51       break;
52     }
53     if( x < p->value )
54       p = p->left;
55     else
```

```
56        p = p->right;
57    }while(1);
58    return;
59  }
60
61  void eliminate(int x){
62    struct vertex *f, *p, *q;
63
64    p = tree;
65    if( p == NULL ) return;
66    do {
67      f = p;
68      if( x < p->value )
69        p = p->left;
70      else if( p->value < x )
71        p = p->right;
72    }while( x != p->value );
73
74    if( p->left == NULL || p->right == NULL ){
75      if( p->right == NULL )
76        q = p->left;
77      else
78        q = p->right;
79      if( p == tree )
80        tree = q;
81      else{
82        if( f->left == p )
83          f->left = q;
84        else
85          f->right = q;
86      }
87    }else{
88      q = p->right;
89      f = q;
90      while( q->left != NULL ){
91        f = q;
92        q = q->left;
93      }
94      p->value = q->value;
95      if( q == f )
96        p->right = q->right;
97      else
98        f->left = q->right;
99    }
100   return;
101 }
102
103 void display(struct vertex *p){
104   if( p == NULL ) return;
105   printf("%d,", p->value );
106   display(p->left);
107   display(p->right);
108   return;
109 }
```