平 成 ２９ 年 度

名古屋大学大学院情報科学研究科
情報システム学専攻
入 学 試 験 問 題

専　　門

平成２８年８月４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．外国人留学生は英語で解答してよい。また、和英辞書などの辞書を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(1)解析・線形代数、(2)確率・統計、(3)プログラミング、(4)計算機理論、(5)ハードウェア、(6)ソフトウェアの６科目がある。(4)~(6)の３科目から少なくとも１科目を選択して解答し、(1)~(3)を含めた６科目から合計３科目を選択して解答せよ。
なお、選択した科目名を解答用紙の指定欄に記入せよ。
(1) (2) (3)の組み合わせを選択した場合は0点となる。

6．解答用紙は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．解答用紙表面に書ききれない場合は、裏面を使用してもよい。
ただし、裏面を使用した場合は、その旨、解答用紙表面右下に明記せよ。

8．解答用紙はホッチキスを外さず、試験終了後に3枚とも提出せよ。

9．問題冊子、草稿用紙は試験終了後に持ち帰ってよい。

解析・線形代数

（解の導出過程を書くこと）

[1] 次の微分方程式について考える．

$$(1+y^2)\,\frac{d^2y}{dx^2} = 2y\left(\frac{dy}{dx}\right)^2 \tag{1}$$

(a) まず，$p = \frac{dy}{dx}$ とおいて，式 (1) の微分方程式を解き，p を y の多項式の形で表せ．ここで，$\frac{dp}{dx} = \frac{dp}{dy}p$ となることに留意せよ．

(b) 次に，(a) で得られた微分方程式において，$y = \tan\theta$ とおくことで，式 (1) の微分方程式の一般解を求めよ．

[2] 漸化式 $a_n = 4a_{n-1} - 3a_{n-2}$ により定義された数列 $\{a_n\}$ について考える．

(a) $\begin{pmatrix} a_n \\ a_{n-1} \end{pmatrix} = A \begin{pmatrix} a_{n-1} \\ a_{n-2} \end{pmatrix}$ を満たす 2×2 行列 A を求めよ．

(b) 行列 A の全ての固有値と，その各々に属する固有ベクトルのうち大きさが 1 のものを求めよ．

(c) 行列 A は，ある行列 P によって，$D = P^{-1}AP$ の形で対角化できる．このような行列 P 及び対角 行列 D を求めよ．

(d) 行列 A^n を求めよ．

(e) 数列 $\{a_n\}$ の一般項 a_n を，a_1 と a_2 を用いて表せ．

[3] 3次元空間中の 領域 $K = \{(x,y,z) \in \Re^3 \,|\, x^2+y^2 \le 1, x \ge 0, y \ge 0, 0 \le z \le \sqrt{2}\}$ 及び平面 $L = \{(x,y,z) \in \Re^3 \,|\, x+y-z = 0\}$ について考える．ここで，$\Re$ は実数全体の 集合 を表す．

(a) xy 平面上の 曲線 $C = \{(x,y) \in \Re^2 \,|\, x^2+y^2 = 1, x \ge 0, y \ge 0\}$ 上の点について，x を用いて，その y 座標を表せ．

(b) 領域 K の中で平面 L と xy 平面に挟まれた 領域 の体積 V を求めよ．

Translations of technical terms

微分 方程式	differential equation	一般項	general term
多項式	polynomial	次元	dimension
一般解	general solution	空間	space
漸化式	recurrence relation	領域	region
数列	series	平面	plane
行列	matrix	実数	real number
固有値	eigenvalue	集合	set
固有ベクトル	eigenvector	曲線	curve
大きさ	magnitude	点	point
対角化	diagonalization	座標	coordinate
対角 行列	diagonal matrix	体積	volume

確率・統計

解の導出過程も書くこと。

1 次のような確率変数X，Yの2次元同時確率分布表があるとき，以下の問いに答えなさい。

X \ Y	1	2	3
1	a	b	c
2	b	c	a
3	c	a	b

(1) a, b, cの間の関係を式で表しなさい。

(2) XとYが互いに独立であるとき，a, b, cの値を求めなさい。

2 次の確率密度関数で表される確率変数Xについて，以下の問いに答えなさい。

$$f_X(x) = \begin{cases} 2\alpha e^{-\lambda x} & (x \geqq 0) \\ 0 & (x < 0) \end{cases}$$

ただし，αとλは定数で，$\alpha > 0$，$\lambda > 0$とする。

(1) αをλで表しなさい。

(2) Xの期待値$E(X)$をλで表しなさい。

(3) Xの分散$V(X)$をλで表しなさい。

3 $(X_1, X_2, ..., X_n)$を，母平均μ，母分散σ^2の母集団における大きさnの標本変量とする。ここで次の統計量を作る。

標本平均
$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

不偏分散
$$U^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2$$

このとき，以下の問いに答えなさい。

(1) 母集団の母数を標本変量の関数値（つまり統計量）と考えることを点推定といい，この関数値を推定量という。そして，推定量に関する性質に一致性と不偏性がある。この2つを説明しなさい。

(2) 上記の$\overline{X}$，U^2はそれぞれμ，σ^2の不偏推定量であることを示しなさい。

用語一覧 (Technical Terms)

確率変数 (random variable),
2次元同時確率分布表 (2-dimensional joint probability distribution table),
互いに独立 (mutually independent), 確率密度関数 (probability density function),
期待値 (expected value), 分散 (variance), 母平均 (population mean),
母分散 (population variance), 母集団 (population), 標本変量 (sample variable),
統計量 (statistics value), 標本平均 (sample mean), 不偏分散 (unbiased variance),
母数 (parameter), 点推定 (point estimation), 推定量 (estimator), 一致性 (consistency),
不偏性 (unbiasedness), 不偏推定量 (unbiased estimator)

プログラミング

プログラム P は，与えられた整数の配列numbers(s 個の要素を持つ)を numbers[0]≦numbers[1]≦…≦numbers[s-1]となるようソートする C言語プログラムである．プログラム P に対して以下の問いに答えよ．

(1) 11, 14 行目の空欄 A, B, C, D にあてはまる式を答えよ．

(2) 29, 30 行目の空欄 E, F, G にあてはまる式を答えよ．

(3) 2 行目の定数 N の定義は 36 行目の配列宣言 numbers[6]の添え字に応じて変更しなければならない場合がある．36 行目の numbers の配列宣言の添え字を m とし，N が最低いくらでなければならないか m を使って答えよ．

(4) プログラム P の実行結果として標準出力に表示される結果を答えよ．

(5) プログラム P の 18 行目のコメント開始記号 "/*"とコメント終了記号"*/"を削除したときのプログラムをプログラム P'とする．P'において，18 行目がはじめて実行されたときに標準出力に出力される実行結果を書け．

(6) プログラム P の 22 行目のコメント開始記号"/*"とコメント終了記号"*/"を削除したときのプログラムをプログラム P''とする．P''において，22 行目が 3 回目に実行されたときに標準出力に出力される実行結果を書け．

プログラム P (行頭の数字は行番号を表す)

```
1   #include<stdio.h>
2   #define N 10
3   void func1(int* numbers, int start, int size) {
4     int h, i, j, k, tmp[N] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
5
6     h = size / 2;
7     i = start;
8     j = start + h;
9     for (k = 0; k < size; k++) {
10      if ((j == start + size) || (i < start + h) && (numbers[i] <= numbers[j])) {
11        [ A ] = [ B ];
12        i++;
13      } else {
14        [ C ] = [ D ];
15        j++;
16      }
17    }
18  /* printf("%d,%d,%d,%d,%d,%d¥n", tmp[0], tmp[1], tmp[2], tmp[3], tmp[4], tmp[5]); */
19    for (k = 0; k < size; k++) {
20      numbers[start + k] = tmp[k];
21    }
22  /* printf("%d,%d,%d,%d,%d,%d¥n", numbers[0], numbers[1], numbers[2], numbers[3],
    numbers[4], numbers[5]); */
23  }
```

```
24    void func2(int* numbers, int start, int size) {
25      int h;
26      printf("%d, %d¥n", start, size);
27      if (size > 1) {
28        h = size / 2;
29        func2(numbers, start, [ E ]);
30        func2(numbers, [ F ], [ G ]);
31        func1(numbers, start, size);
32      }
33    }
34
35    void main(int argc, char** argv) {
36      int numbers[6] = {3, 2, 5, 4, 6, 1 };
37
38      func2(numbers, 0, 6);
39    }
```

Translation of technical terms

プログラム	program	式	expression
整数	integer	定数	constant
配列	array	宣言	declaration
要素	element	添え字	index
ソート	sort	標準出力	standard output
C 言語	C programming language	コメント	comment

計算機理論

[1] $\mathbb{N}$ を 0 以上の整数の全体からなる集合とし，$2^{\mathbb{N}}$ を $\mathbb{N}$ の冪集合，すなわち $\mathbb{N}$ の部分集合の全体からなる集合とする．写像 $F : 2^{\mathbb{N}} \longrightarrow 2^{\mathbb{N}}$ が単調であるとは，任意の $X, Y \subseteq \mathbb{N}$ について $X \subseteq Y$ ならば $F(X) \subseteq F(Y)$ であることを言う．単調な写像 F に対して，$\mathrm{fp}_F \subseteq \mathbb{N}$ を次で定義する．

$$\mathrm{fp}_F = \bigcap \{X \mid X \subseteq \mathbb{N} \text{ かつ } F(X) \subseteq X\}$$

この右辺は，$F(X) \subseteq X$ を満たすような $\mathbb{N}$ の部分集合 X 全ての共通部分を意味する．また，写像 $T : 2^{\mathbb{N}} \longrightarrow 2^{\mathbb{N}}$ を次で定義する．

$$T(X) = \{0\} \cup \{n + 2 \mid n \in X\}$$

(1) T が単調であることを証明せよ．

(2) $\mathbb{N}$ の部分集合 Z は $T(Z) \subseteq Z$ を満たすとする．このとき，全ての 0 以上の偶数 $2m$ は Z の要素であることを，m に関する数学的帰納法によって証明せよ．

(3) $\mathrm{fp}_T = \{n \in \mathbb{N} \mid n \text{ は偶数 }\}$ を証明せよ．

(4) 一般に，単調な写像 $F : 2^{\mathbb{N}} \longrightarrow 2^{\mathbb{N}}$ について (i) $F(\mathrm{fp}_F) \subseteq \mathrm{fp}_F$ と (ii) $F(\mathrm{fp}_F) \supseteq \mathrm{fp}_F$ が成り立つことを証明せよ．

Translation of technical terms

整数	integer	単調である	monotone
集合	set	共通部分	intersection
冪集合	power set	偶数	even number
部分集合	subset	要素	element
写像	mapping	数学的帰納法	mathematical induction

[2] 記号列 w が記号列 w' の接頭辞であるとは，記号列 u が存在し $w' = wu$ を満たすことである．また，記号列 w が記号列 w' の接尾辞であるとは，記号列 u が存在し $w' = uw$ を満たすことである．なお，u は空列でもよいので，記号列 w はそれ自身の接頭辞であり，かつ接尾辞でもある．以下では，アルファベット $\{0,1\}$ 上の下記の言語について考える．

- 010 を接頭辞に持つ記号列すべてからなる言語 L_1
- 010 を接尾辞に持つ記号列すべてからなる言語 L_2

例えば，記号列 01001 は L_1 に含まれるが，00101 は L_1 に含まれない．一方，0010 は L_2 に含まれる．また，01010 は L_1 と L_2 の両方に含まれる．言語 L_1，L_2 について以下の問いに答えよ．

(1) L_1 を認識する決定性有限オートマトンを構成するために必要な 5 つの状態を以下の（ア）～（エ）に $\{0,1\}$ 上の記号列を埋めることで説明せよ．ただし，（ア）～（エ）はすべて異なる記号列である．

- 到達するまでに入力された記号列が［　（ア）　］であり，かつその記号列のみである状態 q_1
- 到達するまでに入力された記号列が［　（イ）　］であり，かつその記号列のみである状態 q_2
- 到達するまでに入力された記号列が［　（ウ）　］であり，かつその記号列のみである状態 q_3
- 到達するまでに入力された記号列の接頭辞が［　（エ）　］である状態 q_4
- 上記以外の状態 q_5

さらに，上記の 5 つの状態のうちで初期状態，最終状態であるべきものがどれであるかを示せ．

(2) L_1 を認識する決定性有限オートマトン A_1 の状態遷移図を描け．なお，解答するオートマトンの状態を 5 つとし，初期状態および最終状態を必ず明記すること．

(3) L_2 を認識する決定性有限オートマトン A_2 の状態遷移図を描け．なお，解答するオートマトンの状態を 4 つとし，初期状態および最終状態を必ず明記すること．

(4) 正規言語のクラスは共通部分を求める演算について閉じているので，$L_1 \cap L_2$ も正規言語である．$L_1 \cap L_2$ を認識する決定性有限オートマトン A_3 の状態遷移図を描け．なお，解答するオートマトンの状態の数は 8 つとし，初期状態および最終状態を必ず明記すること．

Translation of technical terms

記号列	string	到達する	reach
接頭辞	prefix	初期状態	initial state
接尾辞	suffix	最終状態	final state
空列	empty string	状態遷移図	state diagram
アルファベット	alphabet	正規言語	regular language
言語	language	共通部分	intersection
認識する	recognize	演算	operation
決定性有限オートマトン	deterministic finite automaton	閉じている	closed
状態	state		

ハードウェア

[1] RISC（Reduced Instruction Set Computer）とは，プロセッサの命令セットを単純化することで，処理性能の向上を図ったプロセッサのことをいう．RISC の主な特徴として，固定長命令，ロード／ストアアーキテクチャ，遅延分岐方式を挙げることができる．これに関して，次の問いに答えよ．

(1) 命令セットを単純化することで処理性能が向上できる理由を，80 字（英語の場合，35 語）程度で説明せよ．

(2) ロード／ストアアーキテクチャについて，60 字（英語の場合，25 語）程度で説明せよ．

(3) 固定長の命令セットを設計する際の課題の 1 つに，大きい定数値の扱いがある．例えば，32 ビット固定長命令において，32 ビットの定数値をレジスタに代入する処理は，32 ビットでは表現できない．32 ビット固定長命令で，32 ビットの定数値をレジスタに代入する処理を実現する方法 1 つを，80 字（英語の場合，35 語）程度で説明せよ．

(4) 遅延分岐方式の代表的な実装として，分岐命令の直後の命令を，分岐を行う前に実行する方式がある．つまり，分岐を行う場合でも，分岐命令の次の命令が実行される．この方式で処理性能を向上させることができる理由を，80 字（英語の場合，35 語）程度で説明せよ．

[2] IEEE754 方式による単精度2進浮動小数点表現では，図 1 に示すように，全 32 ビットのビット列に対し，符号sは 1 ビット，指数部 e は 8 ビット，仮数部 fは 23 ビットで表現する．指数部は X=127 のゲタばき(バイアス)表現，仮数部は符号絶対値表現により負の数を表現する．表現している数 Nの値は，符号 s，指数部 e，仮数部 fの値に応じて表 1 のようになる．

符号 s（1 ビット）	指数部 e（8 ビット）	仮数部 f（23 ビット）

図 1

表 1

指数部 e	仮数部 f	数 N の値
全ビット 0	全ビット 0	$N = (-1)^s \cdot 0$
	00000000000000000000001～ 11111111111111111111111	$N = (-1)^s \cdot (0.f)_2 \cdot 2^{-(X-1)}$
00000001 ～ 11111110	任意	$N = (-1)^s \cdot (1.f)_2 \cdot 2^{(e)_2-X}$
全ビット 1	全ビット 0	$N = (-1)^s \cdot \infty$
	00000000000000000000001～ 11111111111111111111111	NaN: Nは数ではない

これに関して，次の問いに答えよ．

(1) 10 進数$(804)_{10}$を IEEE754 方式による単精度 2 進浮動小数点表現により表現した場合のビット列を示せ．

(2) IEEE754 方式による単精度 2 進浮動小数点表現により表現できる正の最大値（＋∞を除く）N_{max}を表すビット列を示せ．その上でN_{max}の値を，2^aまたは2^a+2^bまたは2^a-2^b（a，bは整数）の形で答えよ．

(3) IEEE754 方式による単精度 2 進浮動小数点表現により表現できる正の最小値N_{min}を表すビット列を示せ．その上でN_{min}の値を，2^aまたは2^a+2^bまたは2^a-2^b（a，bは整数）の形で答えよ．

[3] OSI（Open Systems Interconnection）の 7 層モデルの 1～5 層および 7 層の名称を述べ，30 字（英語の場合，15 語）程度で特徴を説明せよ．解答は表 2 のような形式で書け．

表 2

層	名前	説明
1		
2		
3		
4		
5		
6	プレゼンテーション層	データの形式，例えば文字コード変換や暗号化・復号などを扱う
7		

Translation of technical terms

プロセッサ	processor	符号	sign
命令セット	instruction set	指数	exponent
処理性能	hardware performance	仮数	fraction
固定長命令	fixed-length instruction	ゲタばき(バイアス)	biased
ロード／ストアアーキテクチャ	load / store architecture	符号絶対値	sign magnitude
遅延分岐	delayed branch	負	negative
定数値	constant value	正	positive
分岐命令	branch instruction	整数	integer
単精度	single precision	層	layer
2 進浮動小数点	binary floating point	暗号化	encryption
表現	representation	復号	decryption
列	sequence		

ソフトウェア

[1] 図1のようなシステム構成を持つ電気ポットを考える．電源を入れることにより，タンクに蓄えられた水をヒータによって加熱する．給湯ボタンを押すと，タンク内の水がポンプによって排出される．電気ポットで沸騰したお湯を利用できる．

加熱制御ソフトウェアは，温度センサと水位センサによって安全な場合に限ってヒータの加熱を指示する．

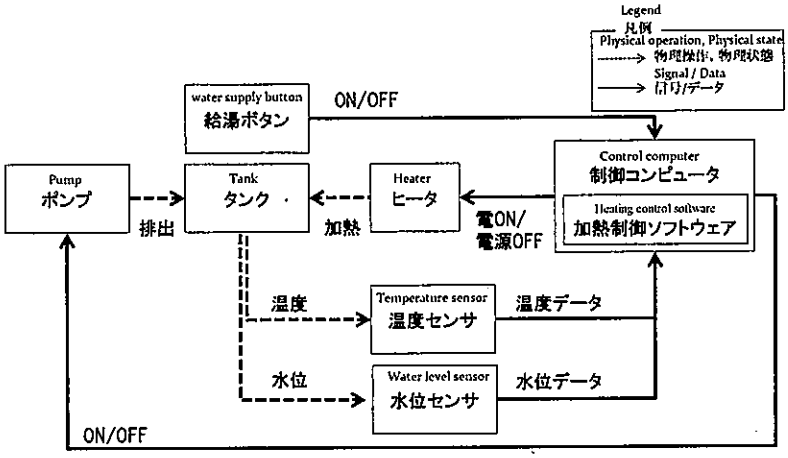


図1　電気ポットのシステム構成

次の安全原則を満たすように，図2に示す加熱制御ソフトウェアの状態図を作成した．

[安全原則1]タンクの水位が下限値以下であるか，上限値以上のとき，加熱してはいけない

[安全原則2]沸騰したら加熱してはいけない

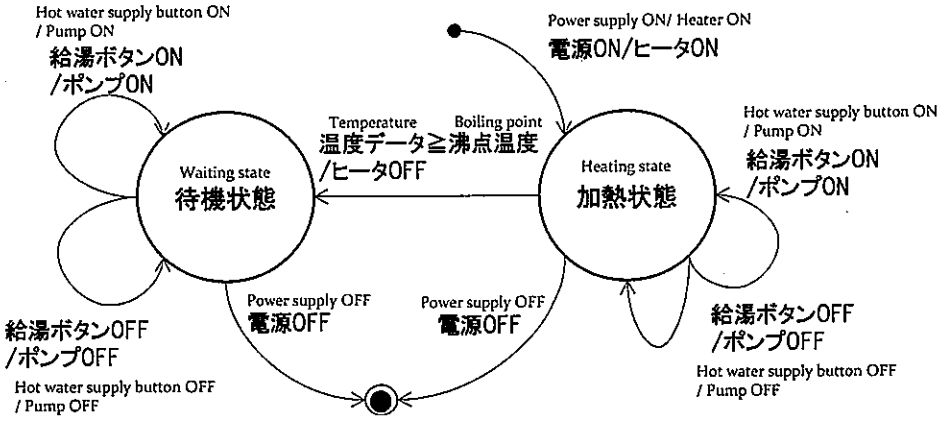


図2　加熱制御ソフトウェアの状態図

ここで，状態遷移のラベルは，記号/の左辺が遷移条件，右辺がアクションを示している．右辺のないラベルはアクションがないことを示している．

このとき，次の問いに答えなさい．

（１）以下の記号を用いて，図２にあるヒータの加熱条件を示しなさい．

記号

V:タンク内の水位，V_L：水位の下限値，V_U:水位の上限値

T：タンク内の水温，T_B：沸点温度

不等号：<

論理記号：&

（２）状態図で示した加熱制御ソフトウェアが加熱安全原則を満たすかどうか解答しなさい．もし，安全原則を満たさないなら，その理由を述べ，安全原則を満たすような，状態図の改善案を述べなさい．

Translation of technical terms

電気ポット	electric pot	下限値	lower bound value
給湯ボタン	hot water supply button	状態遷移	state transition
排出	discharge	遷移条件	transition condition
沸騰した	boiled	水位	water level
加熱制御	heating control	水温	water temperature
状態図	state diagram	沸点温度	boiling temperature
安全原則	safety principle	改善案	improvement plan
上限値	upper bound value		

[2] コンパイラにおける目的コード生成を考える．目的コードはコマンドの列である．コマンドは2種類で，その形式と意味は以下の通りである：

`MOV x y` メモリアドレス y の内容をレジスタ x に転送する．メモリアドレスは変数で表される．

`OP x y` `OP` は二項演算子，x はレジスタ，y はメモリアドレスまたはレジスタであり，x と y に対して演算 `OP` を行い，その結果をレジスタ x に記憶する．

目的コード生成アルゴリズム gen への入力は，1つ以上の演算を含む算術式を表す二分木である（下図参照）．二分木の頂点 v が葉頂点のとき，v の表す変数を `id(v)` と書く．v が内部頂点のとき，v の表す二項演算子を `op(v)`，左の子頂点を `l(v)`, 右の子頂点を `r(v)` と書く．

頂点 v に対して `num(v)` を以下の (i),(ii) で定義する．`num(v)` は，v を根頂点とする部分木が表す算術式（v の表す式と略記）に対して，gen によって生成された目的コードがその値を計算するのに必要なレジスタ数を表す．

(i) v が葉頂点であるとき，

$$\mathtt{num(v)} = \begin{cases} 1 & (\mathtt{v} \text{ がある頂点の左の子頂点であるとき}) \\ 0 & (\mathtt{v} \text{ がある頂点の右の子頂点であるとき}) \end{cases}$$

(ii) v が内部頂点であるとき，`nl=num(l(v))`, `nr=num(r(v))` とすると，

$$\mathtt{num(v)} = \begin{cases} \mathtt{nl} & (\mathtt{nl} > \mathtt{nr} \text{ のとき}) \\ \mathtt{nr} & (\mathtt{nl} < \mathtt{nr} \text{ のとき}) \\ \mathtt{nl}+1 & (\mathtt{nl} = \mathtt{nr} \text{ のとき}) \end{cases}$$

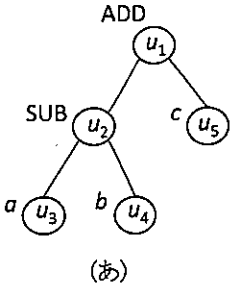
(1) 頂点 v に対して `num(l(v))=1`, `num(r(v))=2` のとき，v の表す式を計算する以下の2つの方法を考える．

(a) `l(v)` の表す式を先に計算してから `r(v)` の表す式を計算する．

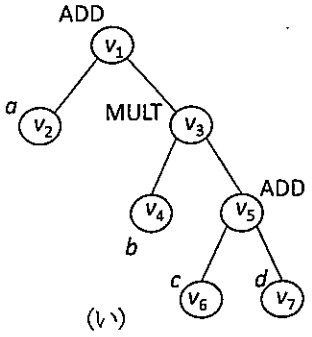
(b) `r(v)` の表す式を先に計算してから `l(v)` の表す式を計算する．

(a) の方法では，`r(v)` の表す式の計算中に，先に計算した `l(v)` の表す式の値を保持するレジスタが1つ必要であるので，レジスタは3つ必要となる．
(b) の方法では，v の表す式を計算するのに必要なレジスタは何個となるかを，100字（英語の場合，40語）程度の説明とともに答えよ．

(2) 下図の二分木 (あ),(い) について，各頂点 v と `num(v)` の値を，$u_1 : 3, u_2 : 1, \cdots$ のように並べて答えよ．



(あ)



(い)

(3) 以下はアルゴリズム gen を疑似コードで記述したものである．疑似コードの行頭の 2 桁は行番号である．
reg はレジスタ名を記憶するための局所変数(きょくしょへんすう)である．
gen の開始時に，利用可能なレジスタ名が，s=[R1,R2,...,Rn] のように，R1 を先頭とするスタック s に格納されている．gen は，与えられた式の計算結果が s の先頭のレジスタに格納されるように，目的コードを生成する．
top(s) はスタック s の先頭要素のレジスタ名を表す（ポップはしない）．
reg = pop(s) は s の先頭要素をポップして reg に代入することを表す．
push(s, reg) は，reg に格納されているレジスタ名を s にプッシュすることを表す．
swap(s) は，スタック s の先頭とその次のレジスタ名を入れ替える．例えば，s=[R1,R2,R3] のとき，swap(s) を実行すると，s=[R2,R1,R3] となる．
print(op, x, y) は，目的コードのコマンド op x, y を生成することを表す．

```
gen(v)
01  char* reg;
02  if ((v は葉頂点) かつ (v はある頂点の左の子頂点))
03    print('MOV', top(s), id(v)) ;
04  else if ( num(r(v))==0 ) {
05    gen(l(v)) ;
06    print(op(v), top(s), id(r(v))) ; }
07  else if ( num(l(v))>=num(r(v)) )  {
08    gen(l(v)) ;
09    reg = pop(s) ;
10    gen(r(v)) ;
11    print(op(v), reg, top(s)) ;
12    push(s, reg) ; }
13  else {
14    swap(s) ;
15    gen(r(v)) ;
16    reg = pop(s) ;
17    gen(l(v)) ;
18    print(op(v), top(s), reg) ;
19    push(s, reg) ;
20    swap(s) ; }
```

(a) 上のアルゴリズムの 07–12 行と 13–20 行とでは，gen の再帰呼出(さいきよびだ)しの順番を変えている．そうすることの利点を 130 字（英語の場合，60 語）程度で述べよ．

(b) 図の 2 分木 (あ),(い) に対して gen を実行したときに生成される目的コードを書け．ただし，スタックの初期値を [R1,R2,R3] とする．

Translation of technical terms

コンパイラ	compiler	頂点	node
目的コード	object code	葉頂点	leaf node
コマンド	command	内部頂点	internal node
メモリアドレス	memory address	子頂点	child node
レジスタ	register	根頂点	root node
二項演算子	binary operator	部分木	subtree
アルゴリズム	algorithm	局所変数	local variable
算術式	arithmetic expression	スタック	stack
二分木	binary tree	再帰呼出し	recursive call