問題 4

除算回路とは，被除数 X を除数 Y で割り算を行い，商 Q を求める回路である．ここで

$$X = Y * Q + R$$

である．

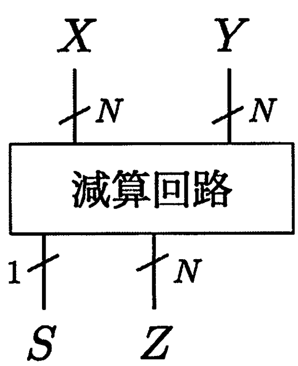
回路を簡単にするため，剰余 R は求めないものとする．また，X, Y は全て正の整数とし，X, Y, Q, R は N ビットで表現されるものとする．

以下の問いに答えよ．

(1) $N = 8$ として，$X - Y$ を計算する減算回路を AND, OR, NOT ゲートを用いて設計せよ．

(2) 問い (1) で求めた減算回路に，計算結果が負であることを示す符号出力 S を追加せよ．

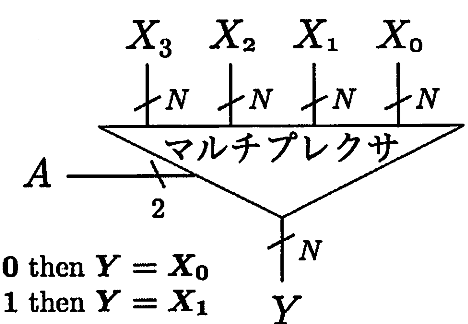
(3) 1 クロックで 1 ビットずつ出力を行う除算回路を減算回路（図 1 のように表記），フリップフロップ，マルチプレクサ（図 2），AND, OR, NOT ゲートを用いて設計せよ．



$Z = X - Y$

if $Z \geq 0$ then $S = 0$ else $S = 1$

図 1．減算回路（N ビット）



if $A = 0$ then $Y = X_0$
if $A = 1$ then $Y = X_1$
if $A = 2$ then $Y = X_2$
if $A = 3$ then $Y = X_3$

図 2．マルチプレクサ（N ビット，4 入力の例）

(4) (3) の除算回路は，1 クロックで 1 ビットずつ出力を行いながら除算を実現している．この除算回路を高速化するため，1 クロックで 2 ビットずつ出力を行う除算回路を設計せよ．

第2問

データ依存と命令レベル並列性に関する以下の設問に答えよ.

(1) 命令レベル並列性に影響を及ぼすデータ依存関係は, フロー依存, 逆依存, 入力依存, および, 出力依存の4種に分類することができる. それぞれどのようなものであるか説明せよ.

(2) (1) で挙げた4種のデータ依存の内, 命令間の先行制約を生じるものをすべて示せ.

(3) データ依存はまた, 真のデータ依存と偽のデータ依存に分けられる. 真のデータ依存とは命令間のデータの授受を表現するもので, 偽のデータ依存は表現しないものである. (2) で答えたデータ依存のそれぞれが, 真偽のいずれのデータ依存であるか示せ.

あるプロセッサのアセンブリコードを下に示す. コード中, L1～L4 はラベルである. r0～r2 はレジスタで, 代入記号 "=" の左辺のレジスタがデスティネーションを, 右辺のレジスタがソースを表す. なお, L1 の ld は, ロード命令で, r0 の内容をアドレスとしてメモリアクセスを行うことを意味する. L2 および L3 の sla は算術左シフト命令である.

```
L1: ld   r1 = *r0;
L2: sla  r2 = r1 << 1;
L3: sla  r1 = r1 << 2;
L4: add  r1 = r1 + r2;
```

(4) このコード中, フロー依存が存在する命令の対をすべて挙げよ. 同様に, 逆依存が存在する命令の対もすべて挙げよ.

(5) 偽のデータ依存を解消する手法を答えよ.

また, 上記コードにその手法を適用した場合, 命令レベル並列性がどのように改善するか説明せよ.

第3問

データベース管理システムについて以下の問いに答えなさい.

（1） ファイルシステムとデータベース管理システムの差異について論ぜよ.

（2） 関係データベースにおけるＳＱＬ言語の特徴について述べよ.

（3） 図書館における図書の貸し出しの為のデータベースについて考える. データベースは次の３つの関係表から構成されるものとする.

利用者（利用者番号, 氏名, 住所, 電話番号),

図書（本番号, 書名, 著者, 出版社),

貸出状況（利用者番号, 本番号, 貸出日).

ここで, 利用者は事前に登録するものとし, 利用者番号が与えられているものとする. 利用者という関係表では利用者の氏名, 住所, 電話番号が管理されている. 図書という関係表は, 本についての情報の管理を行うことを目的としており, 本には本番号という本を一意に識別する番号が付与されているとする. その他, 書名, 著者, 出版社などの情報が格納されている. ただし, １つの本の著者は１名とする.

貸出状況という関係表は誰にどの本をいつ貸し出したかを管理する.

Ａ出版社の本を借りている利用者で, 貸出日から１０日以上経ている利用者の名前と電話番号を列挙するためのＳＱＬを記述せよ. なお, ここで日付型データに対する差演算が用意されているものとする.

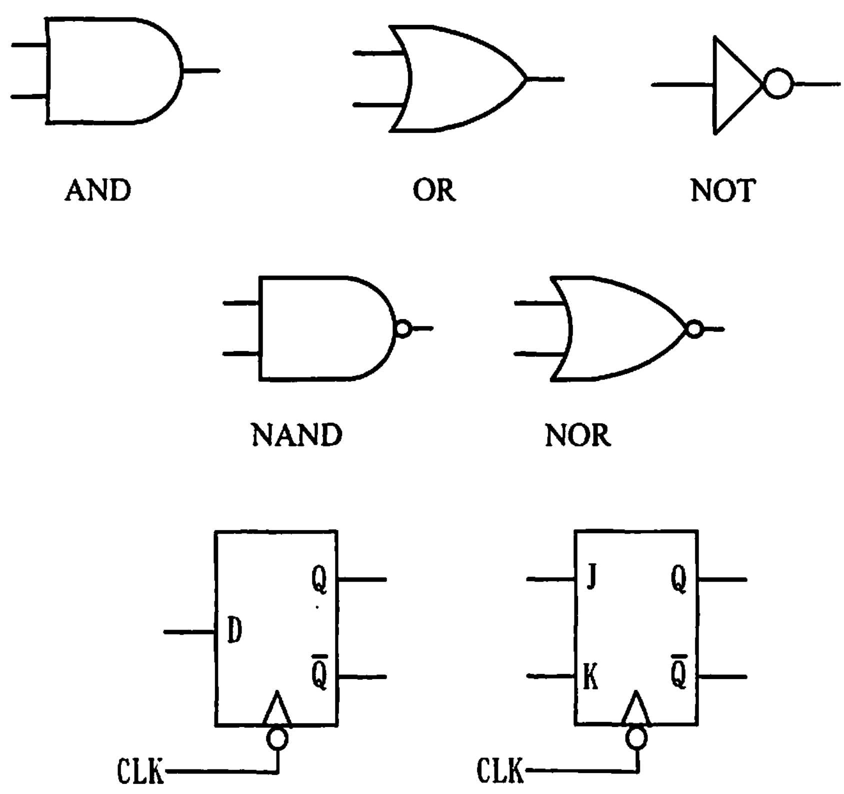
（4） （3）の解を関係代数で表現しなさい. 演算の実行順序と, 問合せの実行に必要とされる時間に関して考察せよ.

（5） 著者毎にその執筆図書の貸出回数を求め, 多い順にならべた表（人気著者リスト）を作成したい. そのためには, どのようなスキーマ変更を行えばよいか. 又, その表を求める為のＳＱＬを記述しなさい.

第2問

6, 1, 0, 2, 5 をこの順に生成する 2 進数の同期型順序回路を設計する．この回路は 5 を生成した次のクロックでは，6 を生成するものとし，クロックが続くかぎり，6, 1, 0, 2, 5 を周期 5 で循環的に生成し続けるものとする．なお，ここで用いてよい論理素子は，AND，OR，NOT，NAND，NOR，D フリップフロップ，JK フリップフロップのうちのどれかとする．

(1) 順序回路とは何か．一行で説明せよ．
(2) 設問の回路を作る準備として，同期式 8 進カウンタの回路を用意する．同期式 8 進カウンタを MIL 記号（図参照）を用いて図示せよ．
(3) (2)をもとに，同期式 5 進カウンタを設計し，MIL 記号を用いて図示せよ．回路はできるだけ簡単化せよ．
(4) (3)をもとに，本題の回路を設計し，MIL 記号を用いて図示せよ．そのさい，(3)の回路をブラックボックスとして使ってよい．なお，カルノー図を使って論理を簡単化すること．



図

第3問

有向グラフに関する以下の設問に答えよ．本問題では，有向グラフは自己ループも多重辺も含まないものとする．

(1) 図1に示された有向グラフの隣接行列による表現と，隣接リストによる表現を図示せよ．隣接リストの表現としては図2に示すものを用いよ．

(2) 隣接行列の各要素，隣接リストにおける各頂点のラベル，および各ポインタを格納するのに同じ大きさのメモリを使用する場合，図 1 の有向グラフについてどちらの表現がよりメモリ容量を必要とするか答えよ．

(3) 有向グラフの頂点数をNに固定したとき，どちらの表現でもメモリ容量が同じになる辺の数を答えよ． この条件を満たすとき有向グラフは必ず弱連結となるかどうか，理由と共に答えよ．ただし，弱連結とは有向辺を無向に置き換えた無向グラフが連結であることを言う．

(4) 以下の疑似コードは有向グラフ G を深さ優先探索するものである．G は頂点集合 V と有向辺集合 E からなるものとする．図 1 の有向グラフを頂点 0 から深さ優先探索したときの探索木を図示せよ．

```
Traverse(Graph G)
   foreach v ∈ V { v.setMark(UNVISITED) }
   foreach v ∈ V { if (v.getMark() == UNVISITED) DFS(G, v) }
DFS(Graph G, Vertex v)
   v.setMark(VISITED)
   foreach (v, w) ∈ E {
      if (w.getMark() == UNVISITED) DFS(G, w)
   }
```

(5) グラフ上において，始点と終点が同じである有向経路を有向サイクルと呼ぶ．グラフに有向サイクルが存在すればそのうち 1 つを発見し，それに含まれる頂点の集合を出力するアルゴリズムを，疑似コードを用いて記述せよ．

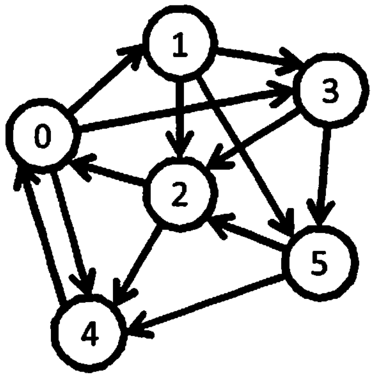


図1

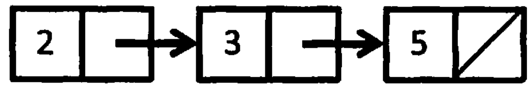


図2

第4問

複数の無線端末が同一のアクセスポイントにアクセスする無線通信システムにおいては，多数の端末が同時に接続すると，そのスループットが劣化する．このため，最大同時接続数を制限してスループット劣化を防止することを試みる．

(1) スループットの劣化が起こる理由を定性的に示せ．

(2) 空港ターミナルのような場所では十分多数の端末が存在するので，接続はポアソン生起すると仮定できる．今，接続が生起率 2 [1/hour]でポアソン生起し，接続時間は平均 1 [hour]の指数分布に従うものとする．最大同時接続数を制限しないものとした場合の同時接続数の確率分布と平均同時接続数を求めよ．

(3) (2)において，最大同時接続数を 3 とした場合に接続に失敗する確率を求めよ．

(4) 次に，屋内会議室で利用する場合を想定して，端末数は有限であるとする．個々の端末は接続状態と無接続状態を交互に繰り返し，接続時間，無接続時間共に平均 1 [hour]の指数分布に従うものとする．それぞれの端末は独立にふるまうものとし，また接続に失敗した端末は再び上記の指数分布に従う無接続状態となるものとする．最大同時接続数を制限しない場合に，端末が接続状態にある確率を求めよ．また端末数が 4 台の場合に，同時接続数の確率分布と平均同時接続数を求めよ．また平均同時接続数が(2)の場合と一致することを示せ．

(5) (4)において，最大同時接続数を 4 とした場合に接続に失敗する確率を示せ．同様に(4)において，最大同時接続数を 3 とした場合に接続に失敗する確率を求め，それが(3)の場合に比べ小さくなる理由を説明せよ．

(6) 接続端末数が大きい場合にはアクセスポイントを増設することが考えられる．この際の留意点について「干渉」，「チャネル」の 2 つの用語を用いて簡単に記せ．

第5問

一定の時間間隔で，赤（R），緑（G），青（B）のいずれかのランプの光る装置がある．その光り方には，以下のような特徴がある．

- Rの直後には，RとGのどちらかが，それぞれ0.8と0.2の確率で光る．
- Gの直後には，GとBのどちらかが，それぞれ0.9と0.1の確率で光る．
- Bの直後には，BとRのどちらかが，それぞれ0.9と0.1の確率で光る．

この装置の光の点灯を情報源Sとして，以下の問いに答えよ．

(1) 情報源Sは，一次のマルコフ情報源である．その状態遷移図を示せ．

(2) R, G, B それぞれが点灯する定常状態確率を求めよ．

(3) 情報源Sからの点灯系列を効率よく符号化して伝送したい．点灯を2つずつまとめて0,1からなる2元符号を割り当てる場合に，最も効率の良い符号を設計せよ．また，その時の1点灯あたりの平均符号長を求めよ．

(4) 同じ色の点灯が継続して続く傾向があることを考慮し，以下に示す9つの点灯のまとまりに対して0,1からなる2元符号の割り当てを行う場合，最も効率の良い符号を設計せよ．また，その時の1点灯あたりの平均符号長を求めよ．

RG,　RRG,　RRR,　GB,　GGB,　GGG,　BR,　BBR,　BBB

(5) この情報源Sの平均符号長はどこまで短くすることができるか．その最小値を求めよ．

（なお，計算に当たっては，$\log_2 3=1.58$　$\log_2 5=2.32$ を必要に応じて用いよ）

第2問

仮想記憶に関する以下の問いに答えよ．

(1) 参照の局所性は，空間局所性と時間局所性に分類される．それぞれどのようなものであるか説明せよ．

(2) ページテーブルとは何か説明せよ．

(3) Translation Lookaside Buffer (TLB) とは何か説明せよ．また，TLB ミスとページテーブルの関係について説明せよ．

(4) あるロード命令の実行中にページフォールトが発生したとする．この命令のアドレス計算から，ページフォールトの処理の後にこの命令の実行が再開されるまでの過程を説明せよ．

(5) プロセッサによってはTLB ミスの処理をハードウェアで行うことがある．一方で，ページフォールトの処理をハードウェアのみで行うことは例外的である．このような違いが生じる理由について論ぜよ．

第3問

$f(0)=0, f(1)=1$ および漸化式 $f(n)=f(n-1)+f(n-2)$ $(n>1)$ で表される数列はフィボナッチ数列と呼ばれる．これについて以下の問いに答えよ．

(1) 漸化式に従い，再帰呼び出しを使って $f(n)$ を計算するプログラムを擬似コードにより記述せよ．

(2) 漸化式に従い，再帰呼び出しを使わずに $f(n)$ を計算するプログラムを擬似コードにより記述せよ．

(3) 64 ビット整数を用いて計算する場合，(1) および (2) それぞれの方法の問題点を述べよ．

(4) フィボナッチ数列の一般項は

$$f(n)=\frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^n-\left(\frac{1-\sqrt{5}}{2}\right)^n\right)$$

となることが分かっている．この式に従って浮動小数点数を用いて計算する場合，(2) の方法と比較し利害得失を述べよ．

第２問

以下の問いに答えよ．

(1) 現代的なスーパースカラプロセッサは，プログラムのメモリアクセス性能をあげるために以下を含むいくつもの機構を備えている．

(a) キャッシュ

(b) ハードウェアによる先読み

(c) ノンブロッキングキャッシュ

それぞれの機構が何をするか，およびそれぞれがメモリアクセス性能をあげるためにどう貢献するのかを簡潔に説明せよ．

(2) ある構造体の配列および，その全要素にアクセスするプログラムを考え，アクセスの方法や配列の大きさが性能に与える影響について考察する．具体的には，以下の方法 1〜4 を考える．

方法 1: 全要素を逐次的に，配列の先頭から終わりまでアクセスする．

方法 2: 配列の添字を乱数で生成し，その添字の要素に，それらが生成された順でアクセスする．配列の全ての添字がちょうど一度ずつ生成されるとする．なお，乱数ひとつを生成するのにかかる時間は 7〜8 サイクル程度で，メモリアクセス命令は必要ない．

方法 3: 各要素に後続の要素を指すポインタを持たせ，全要素をひとつの線型リストにする．そして，そのポインタをたどりながらアクセスする．ただし，ある要素は，配列上の直後の要素を指すようにする．結果として要素は方法 1 と同じ順序でアクセスされる．

方法 4: 方法 3 と同様だが，ポインタは方法 2 と同じ乱数で生成される．つまり各要素のポインタは，方法 2 においてその要素の次にアクセスされる要素を指す．結果として要素は方法 2 と同じ順序でアクセスされる．

一要素は 16 バイトとする．プロセッサはレベル 1 から 3 までのキャッシュを持ち，大きさはそれぞれ，32KB, 256KB, 4MB であるとする (1KB $= 2^{10}$ バイト, 1MB$= 2^{20}$ バイト)．

それぞれの方法で，同じ配列を繰り返し続けざまに何度もアクセスし，一アクセスあたりの平均時間を測定した．その時間は方法および，配列の要素数 (N) に応じて変化し，以下の表のようになった．数字はアクセス一回あたりの平均時間 (プロセッササイクル数) を示す．

	$N \approx 2^{10}$	$N \approx 2^{22}$
方法 1	1.1	8.0
方法 2	8.4	47.5
方法 3	7.0	(x)
方法 4	7.0	199.2

以下のそれぞれの場合に，2 つの性能の違いに最も関係の深いプロセッサの機構は何か? 上記の (a) から (c) の中から選んで答えよ．どれも関係しない場合は，なしと答えよ．根拠も示せ．

(2-1) (方法 1, $N \approx 2^{10}$) と (方法 2, $N \approx 2^{10}$)

(2-2) (方法 4, $N \approx 2^{10}$) と (方法 4, $N \approx 2^{22}$)

(2-3) (方法 1, $N \approx 2^{22}$) と (方法 2, $N \approx 2^{22}$)

(2-4) (方法 2, $N \approx 2^{22}$) と (方法 4, $N \approx 2^{22}$)

(3) 表中の (x) の値はどの程度か? 以下のうちから最もありそうな値を選び，理由を述べよ.

(a) 8.0 付近. つまり，方法 1 と同程度.

(b) 47.5 付近. つまり，方法 2 と同程度.

(c) 199.2 付近. つまり，方法 4 と同程度.

第３問

変数 x に関する多項式を $f = a_0 + a_1x + \cdots + a_mx^m$，$g = b_0 + b_1x + \cdots + b_nx^n$（$a_i, b_i$ は実数．$a_m \neq 0$，$b_n \neq 0$）とする．多項式 f, g の先頭項を $\mathrm{LT}(f) = a_mx^m, \mathrm{LT}(g) = b_nx^n$ とし，多項式 f と g の次数を $\deg(f) = m, \deg(g) = n$ と表す．多項式 f を，0 でない多項式 g で割る除算は以下で与えられる．

$$f = qg + r.$$

ここで商 q，余り r は x に関する多項式で，$r = 0$ または $\deg(r) < \deg(g)$ が成り立つ．この場合 $r = \mathrm{remainder}(f, g)$，$q = \mathrm{quotient}(f, g)$ と表す．

(1) $f = x^2 + 7x + 3$，$g = x + 1$ の場合に $\mathrm{quotient}(f, g)$ と $\mathrm{remainder}(f, g)$ を求めよ．

(2) $q = \mathrm{quotient}(f, g)$ と $r = \mathrm{remainder}(f, g)$ を計算する除算アルゴリズムの擬似コードを（a）を埋めることで完成させよ．ただし，単項式（一つの項だけからできている式．例：$7x^3$ や $-5x^{10}$）の四則演算及び多項式の加法・減法はそのまま使えるとして良い．

```
Input : f, g
Output : q, r
q = 0, r = f
while (r ≠ 0 and deg(g) ≤ deg(r)) {
    q = q + LT(r)/LT(g)
    r = ______(a)______
}
```

(3) (2) で示した除算アルゴリズムは必ず停止することを示せ．

(4) 多項式 f，g の最大公約元 GCD（Greatest Common Divisor）とは，以下の条件を満たす多項式 h を表す．

- h は f と g を割り切る
- 多項式 p が f と g を割り切るなら，p は h を割り切る

この場合 $h = \mathrm{GCD}(f, g)$ と表す．$\mathrm{GCD}(f, g)$ は 0 でない定数倍の違いを除いて一意に決まる．$f = qg + r$ の時，$\mathrm{GCD}(f, g) = \mathrm{GCD}(f - qg, g)$ 及び $\mathrm{GCD}(f, 0) = f$ の関係式を用いると，$\mathrm{GCD}(f, g)$ は以下のコードで計算できる（一般性を失うことなく $\deg(f) \geq \deg(g)$ を仮定する）．空欄（b）と（c）を埋めよ．

```
Input f, g
Output: h
h = f
s = g
while (s ≠ 0) {
    rem = remainder(h, s)
    h = ______(b)______
    s = ______(c)______
}
```

(5) 任意の f と g（$\deg(f) \geq \deg(g)$）に対して，GCD(f, g) を実行したとき GCD アルゴリズム中の while ループにある remainder が呼ばれる回数の上界を計算せよ．また，結果の理由も書け．

第4問

ノードとリンクから構成されるネットワークにおける IP パケットの配送経路を計算するために以下に述べるアルゴリズムが適用されたシステムを考える．

経路計算アルゴリズム:

各ノードは，各ノードがリンクで接続されているすべての隣接ノードに，{ 宛先ノード，次ホップノード，ホップ数 } を行ベクトルとする経路表を 30 [sec] ごとに通知する．図 1 は，ノード A の経路表の例を示している．なお，図のホップ数 $d(i,j)$ は，次の計算式にしたがって計算され，自ノード i から宛先ノード j に到達するために必要な最小ホップ数を示している．

$$d(i,j) = \min\{d(i,k) + d(k,j)\}, \qquad k \text{ はノード } i \text{ のすべての隣接ノード}$$

なお，同じコストの経路が存在する時には，ノードの番号がより小さい値を持つ隣接ノードを経由する経路が選択されるものとする．

以下の問いに答えよ．

(1) 図 2 のネットワークにおいて，経路表の交換がノード間で十分に行われ，経路表が収束した時のノード 1 の経路表を示せ．図中の○ (丸) がノードを表しその中の数字がノード番号を示しているものとする．ノードを接続するリンクは線で示されており Li (i は整数) でリンクを表現している．

(2) 各ノードの経路表の情報から，ノード 1 を根とする残りのすべてのノード (2, 3, 4, 5) への転送経路を示す Spanning Tree を作成することができる． この Spanning Tree を示せ．

(3) リンク L1 と L8 が同時に切断された． 経路表が収束した時の，ノード 1 の経路表を示すとともに，収束時のノード 1 を根とする残りのすべてのノード (2, 3, 4, 5) への転送経路を示す Spanning Tree を示せ．

(4) 図 3 に示すように，ノード 3 と同じノード番号を持つノード 3a が，リンク L10 を用いてノード 1 と，リンク L9 を用いてノード 5 と接続された． この新しく接続されたノード 3a は，図の右端のノード 3 と同じノード番号を用いて経路表を隣接ノードに通知するものとする． 経路表の交換がノード間で十分に行われ，経路表が収束した時の，ノード 3 を根とする Spanning Tree とノード 3a を根とする Spanning Tree をそれぞれ示せ．

(5) 同じ番号を持つ複数のノードを，意図的にインターネット上に存在させることがある．この運用法の良い利用法と悪い利用法を示せ．

宛先ノード	次ホップノード	ホップ数
A	A	$d(A,A) = 0$
B	C	$d(A,B) = 3$
C	C	$d(A,C) = 1$
⋮	⋮	⋮
Z	B	$d(A,Z) = 4$

図 1

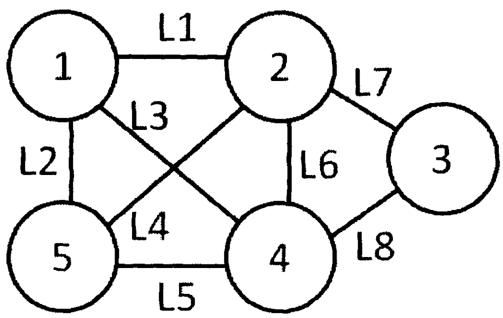


図 2

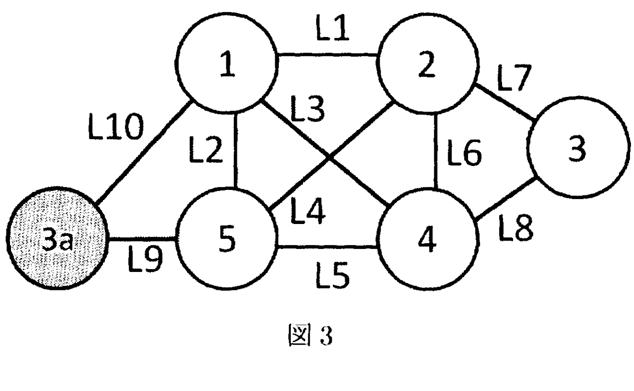
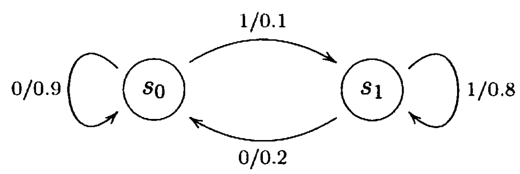


図 3

第５問

以下のような状態遷移図で示される二元単純マルコフ情報源を考える．



以下の問いに答えよ．

(1) 状態 s_0, s_1 の定常確率 w_0, w_1 を求めよ．

(2) 定常状態において出力 1 の発生する確率を求めよ．

(3) 出力系列から，直前および直後が 0 である 1 の連続 (1 のラン) を任意に一つ取り出した時，その長さが 1, 2, k である確率をそれぞれ求めよ．

(4) 1 のランの平均長を求めよ．

(5) この情報源のエントロピーを求めよ．

(6) (2) で求めた生成確率でランダムに 1 が発生する情報源のエントロピーを求めよ．この値と (5) で求めた値の違いについて論ぜよ．

(なお，計算に当たっては，$\log_2 3 = 1.58$, $\log_2 5 = 2.32$ を必要に応じて用いよ.)

問題 2

n 個の実数からなる数列 $A[1..n] = (a_1, a_2, \ldots, a_n)$ を考える．ただし $n \geq 2$ とする．このとき，A の部分数列 $A[i..j] = (a_i, a_{i+1}, \ldots, a_j)$ 中の最大値を $MAX(i, j)$ と表す．

以下の問いに答えよ．

(1) $1 \leq i \leq n-1$ を満たすすべての整数 i に対する $MAX(i, i+1)$ の値を計算する $O(n)$ 時間のアルゴリズム (I) を示せ．また $1 \leq i \leq n - n/2 + 1$ を満たすすべての整数 i に対する $MAX(i, i + n/2 - 1)$ の値を計算する $O(n^2)$ 時間のアルゴリズム (II) を示せ．

(2) r を $1 \leq r \leq n/2$ を満たす整数であるとする．$1 \leq i \leq n - r + 1$ を満たすすべての i に対し $MAX(i, i + r - 1)$ の値をすでに計算し，それらの値を保持しているとする．このとき，$1 \leq \ell \leq m \leq n$ および $r \leq m - \ell < 2r$ を満たす任意の整数の組 ℓ, m に対し，$MAX(\ell, m)$ を $O(1)$ 時間で計算するアルゴリズムを示せ．

(3) $1 \leq i \leq n - 2^k + 1$, $k \geq 0$ を満たす整数の組 i, k に対し，$MAX(i, i + 2^k - 1)$ の値をすでに計算し，それらの値をテーブル T に保持しているとする．このとき，$1 \leq i \leq j \leq n$ を満たす任意の整数の組 i, j に対し，$MAX(i, j)$ を T を用いて $O(1)$ 時間で計算するアルゴリズムを示せ．

(4) 問い (3) で述べたテーブル T を求めるアルゴリズムを示し，その時間計算量を示せ．

(5) 数列 $A[1..n]$ 上で，$1 \leq i \leq j \leq n$ を満たす任意の i, j に対し $MAX(i, j)$ を $O(\log n)$ 時間で求めることを可能とするような，数列 $A[1..n]$ に対する $O(n)$ 時間の前処理アルゴリズムを示せ．

問題 4

2 つの n ビット符号なし整数の加算を行うハードウェアを考える．入力を A, B とし，それぞれの第 i ビット $(i = 0, 1, \ldots, n-1)$ を A_i, B_i と書く．和は $n+1$ ビット整数 S とし，第 i ビット $(i = 0, 1, \ldots, n)$ を S_i と書く．すなわち

$$A = \sum_{i=0}^{n-1} 2^i A_i, \qquad B = \sum_{i=0}^{n-1} 2^i B_i, \qquad S = \sum_{i=0}^{n} 2^i S_i = \sum_{i=0}^{n-1} 2^i (A_i + B_i)$$

である．以下の問いに答えよ．

なお，問い (3), (4), (5) においては，論理回路は AND, OR, NOT, XOR ゲートを用いて設計するものとし，導出の説明なしに回路図のみ示せ．

(1) 以下の符号なし 2 進数の加算を実行せよ．

$$0011000110100111 + 0110011001011010 \qquad [1]$$

(2) $0 \leq i \leq j \leq n-1$ を満たす整数の組 i, j に対して，部分和 $S_{i,j}$ を以下のように定義する．

$$S_{i,j} = \sum_{k=i}^{j} 2^k (A_k + B_k)$$

また，キャリー伝搬シグナル $P_{i,j}$ とキャリー生成シグナル $G_{i,j}$ を以下のように定義する．

$$P_{i,j} = \begin{cases} 1 & \text{if } S_{i,j} = 2^{j+1} - 2^i \\ 0 & \text{otherwise} \end{cases}$$

$$G_{i,j} = \begin{cases} 1 & \text{if } S_{i,j} \geq 2^{j+1} \\ 0 & \text{otherwise} \end{cases}$$

式 [1] の加算における $P_{0,1}$, $G_{0,1}$, $P_{2,3}$, $G_{2,3}$, $P_{0,3}$, $G_{0,3}$ を求めよ．

(3) i は $0 \leq i \leq n-1$ を満たす整数であるとする．A_i, B_i を入力として $P_{i,i}$, $G_{i,i}$ を出力する論理回路を示せ．

(4) i, j, k は $0 \leq i < j \leq k \leq n-1$ を満たす整数であるとする．$P_{i,j-1}$, $G_{i,j-1}$, $P_{j,k}$, $G_{j,k}$ を入力として，$P_{i,k}$ と $G_{i,k}$ を出力する論理回路を示せ．

(5) i は $1 \leq i \leq n-1$ を満たす整数であるとする．$G_{0,i-1}$, A_i, B_i を入力として，S_i を出力する論理回路を示せ．

(6) 問い (3), (4), (5) で設計した論理回路を用いて，2 つの n ビット符号なし整数の加算を行う論理回路の構成方法を論じよ．ただし，ファン・インがある定数以下の論理ゲートのみを用いて，論理ゲートの段数が $O(\log n)$，論理ゲートの個数が $O(n)$ となるようにせよ．

問題 2

自然数 $i \in [0, 2^n - 1]$ の n ビットによる表現を考えると，通常の 2 進数としてのビット表現 $(a_i)_{i\in[0,2^n-1]}$ 以外にもさまざまなものが考えられる．それらのうち，次の条件 A を満たすようなビット表現 $(b_i)_{i\in[0,2^n-1]}$ を考える．

条件 A：
- $b_0 = \mathbf{0}$ である．
- 各 $i \in [0, 2^n - 2]$ に対して，b_i と b_{i+1} とを比較すると 1 ビット異なる．
- b_{2^n-1} と b_0 とを比較した場合も 1 ビット異なる．
- $i, j \in [0, 2^n - 1]$ に対して，$i \neq j$ なら $b_i \neq b_j$ である．

ここで $\mathbf{0}$ は n 桁のビット列 $00\ldots0$ を表す．

たとえば $n = 2$ のとき，通常の 2 進数としてのビット表現 $(a_i)_{i\in[0,3]}$ は左下のようになる．また，条件 A を満たすビット表現 $(b_i)_{i\in[0,3]}$ は右下の 2 種類しかない．

a_0	a_1	a_2	a_3
00	01	10	11

b_0	b_1	b_2	b_3
00	01	11	10
00	10	11	01

以下の問いに答えよ．

(1) $n = 3$ のとき，条件 A を満たすビット表現 $(b_i)_{i\in[0,7]}$ は何種類あるか．

(2) 条件 A をみたすビット表現 $(b_i)_{i\in[0,2^n-1]}$ のうちのある一種類については，通常の 2 進ビット表現 $(a_i)_{i\in[0,2^n-1]}$ から以下の手続きで計算できる．

$$b_i \ = \ a_i \ \boxed{\text{i}} \ (\ a_i \ \boxed{\text{ii}} \ 1\ )$$

ただし空欄 $\boxed{\text{i}}$, $\boxed{\text{ii}}$ にはそれぞれ以下の表で示す演算子のいずれかが入る．これらの空欄に当てはまる演算子を答えよ．

演算子	説明
`a	b`	a と b のビットごと論理和．
`a & b`	a と b のビットごと論理積．
`a ^ b`	a と b のビットごと排他的論理和．
`a << x`	a を左に x ビットシフトさせる．右端には 0 を埋める．
`a >> x`	a を右に x ビットシフトさせる．左端には 0 を埋める．

(3) 問い (2) の手続きによって計算されるビット表現 $(b_i)_{i\in[0,2^n-1]}$ が，確かに条件 A を満たすことを示せ．

(4) 逆に，問い (2) で得られたビット表現 $(b_i)_{i\in[0,2^n-1]}$ から $(a_i)_{i\in[0,2^n-1]}$ を計算する手続きを以下のように表すことができる．ただし空欄 $\boxed{\text{iii}}$, $\boxed{\text{iv}}$ にはそれぞれ問い (2) の表に示した演算子のいずれかが入る．これらの空欄に当てはまる演算子を答えよ．

```
c = 0;
for (d = b_i; d != 0; d = d [iii] 1){
      c = c [iv] d;
}
return c;
```

問題 3

$\mathbf{N} = \{0, 1, 2, \ldots\}$ を非負整数の全体，x を $\mathbf{N}$ 上を動く変数，$b(x)$ および $f(x)$ を $\mathbf{N}$ から $\mathbf{N}$ への帰納的な（すなわち計算可能な）全域関数，P を次のプログラムとする．

```
while  b(x) = 0  do  {
   x := f(x);
}
return  x;
```

任意の部分集合 $S \subseteq \mathbf{N}$ に対して，$\mathrm{wp}(S) \subseteq \mathbf{N}$ を次のような集合と定義する．

$$\left\{ n \in \mathbf{N} \,\middle|\, \begin{array}{l} \text{初期値 } x = n \text{ のもとで } P \text{ を実行すると，} \\ \text{停止して戻り値が } S \text{ に属する} \end{array} \right\}$$

以下の問いに答えよ．

(1) $b(x) = x\%2$ (x を 2 で割ったときの余り)，$f(x) = x/2$ (x を 2 で割ったときの商) としたとき，$\mathrm{wp}(\{0, 1\})$ は何になるか．また，$\mathrm{wp}(\mathbf{N})$ は何か．(導出過程は不要)

(2) $n \in \mathrm{wp}(S)$ かつ $b(n) \neq 0$ ならば $n \in S$ であることを示せ．

(3) $n \in \mathrm{wp}(S)$ かつ $b(n) = 0$ ならば $f(n) \in \mathrm{wp}(S)$ であることを示せ．

(4) 集合 $I \subseteq \mathbf{N}$ が以下の条件を満たすならば，$I \subseteq \mathrm{wp}(S) \cup (\mathbf{N} \setminus \mathrm{wp}(\mathbf{N}))$ が成り立つことを示せ．

- ($n \in I$ かつ $b(n) \neq 0$) ならば $n \in S$
- ($n \in I$ かつ $b(n) = 0$) ならば $f(n) \in I$

(5) $\mathrm{wp}(\mathbf{N})$ が帰納的に可算であることを説明せよ．

(6) $\mathrm{wp}(\mathbf{N})$ が帰納的でないような，帰納的な全域関数 $b(x)$ と $f(x)$ が存在することを説明せよ．

問題 4

10^{10} bps を有する物理ネットワーク層を仮定する．物理ネットワーク層は，以下のフォーマットで定義される 1250 バイトの固定長パケットを送受する．DEST，SRC は，それぞれ受信先，送信元のアドレスである．DATA 領域にはアプリケーションプログラムのデータが格納される．CRC は物理ネットワーク層において計算される Cyclic Redundancy Check である．

```
+---------------+-------------+-----------------+-------------+
| DEST (4bytes) | SRC (4bytes) | DATA (1238bytes) | CRC (4bytes) |
+---------------+-------------+-----------------+-------------+
```

本物理ネットワーク層を利用する API が以下のように提供されている．

- `void netwrite (void *pckt);`
  pckt で示される固定長パケットを送信する．宛先は固定長パケット内の DEST に従う．パケット長をバンド幅で除した時間で送信処理を行い，その後直ちに戻る．

- `void netread (void *pckt);`
  固定長パケットを受信し，pckt で示されるメモリ領域に格納する．パケット全体が到着すると，直ちに戻る．

物理ネットワーク層の通信は非同期であるとする．また常にエラーなくパケットの送受信がされるとする．以下の問いに答えよ．

(1) まず，ソフトウェアのオーバヘッドと通信遅延 (1 つの固定長パケットが送信側コンピュータから送信開始されてから受信側コンピュータで受信開始されるまでの遅延) は無視できるほど小さいと仮定して，ユーザレベルにおけるデータ転送スループットの理論最大値を計算せよ．

(2) 次に，通信遅延を 4 μs で，ソフトウェアオーバーヘッドは無視できるとする．

コンピュータ A は netwrite を用いてコンピュータ B に 1 パケットを送信し，即座にコンピュータ B からのパケットを受信するため netread を呼ぶ．コンピュータ B は十分早く netread を呼び出しており，コンピュータ A からのパケットを受信したら，即座に，あらかじめ準備しておいた別の 1 パケットを netwrite を用いてコンピュータ A に送信する．

このとき，コンピュータ A における，netwrite の呼び出しから netread のリターンまでの所要時間を求めよ．なお，他のプロセスや他のコンピュータが通信している可能性は考えないものとする．

(3) 以下のプロトタイプに従い，send 関数を C 言語で書け．本関数は data が指すメモリアドレスから size バイトのデータを netwrite を用いて dest で示される宛先に送信する．ただし最初のパケットでデータ長 size も一緒に送ること．また netwrite の引数には別途 1250 バイトのバッファとして確保されている void*型の大域変数 pcktarea を用いなければならない．

`void send (NETADDR dest, void *data, int size);`

自分のアドレスは NETADDR 型の大域変数 myaddr に格納されているとする．また，他のプロセスや他のコンピュータが通信している可能性は考えないものとする．int 型と NETADDR 型は 4 バイト，char 型は 1 バイトとする．

(4) 問い (3) において，送信側コンピュータのメモリコピーのバンド幅が低いと，データ転送スループットが低下することを説明せよ．

問題 1

以下では正の整数 n に対し

$$\omega_n = \exp(-2\pi i/n) = \cos(2\pi/n) - i\sin(2\pi/n)$$

とする．また，行列 F_n は (j,k) 要素が

$$\omega_n^{jk} \ = \ \overbrace{\omega_n \cdot \omega_n \cdot \cdots \cdot \omega_n}^{jk \text{ times}}$$

であるような $n \times n$ 行列とする．ただし添字は $1 \leq j,k \leq n$ を動く．I_n は n 次の単位行列，O_n はすべての要素が 0 であるような $n \times n$ の行列とする．行列 A に対して $A^\top$ は A の転置を表す．以下の問いに答えよ．

(1) m が正の整数のとき， $\omega_{4m}^{m}, \omega_{4m}^{2m}, \omega_{4m}^{3m}, \omega_{4m}^{4m}$ を求めよ．

(2) $n \times 2n$ の行列 $J_n = (I_n \ O_n)$ を考える．$\omega_{2n}^{2k} = \omega_n^k$ に注意して，$J_n F_{2n} P_n = F_n$ となるような P_n を求めよ．ただし P_n は $2n \times n$ の行列で，各列はひとつの 1 と $2n-1$ 個の 0 からなる．

(3) P_n の要素をひとつ上にずらしたものを Q_n とする．すなわち

$$(Q_n)_{jk} = \begin{cases} 0 & \text{for } j = 2n \\ (P_n)_{(j+1)k} & \text{otherwise} \end{cases}$$

である．このとき $(P_n \ Q_n)^\top (P_n \ Q_n) = I_{2n}$ となることを示せ．

(4) $\Pi_n = (P_n \ Q_n)$ とする．

$$F_{2n}\Pi_n = \begin{pmatrix} F_n & A_n F_n \\ B_n F_n & C_n F_n \end{pmatrix} = \begin{pmatrix} I_n & A_n \\ B_n & C_n \end{pmatrix} \begin{pmatrix} F_n & O_n \\ O_n & F_n \end{pmatrix}$$

となるような A_n, B_n, C_n を求めよ．また，A_n, B_n, C_n の非零要素数はそれぞれいくつか．

(5) $\boldsymbol{x}$ が $2n$ 次ベクトルのとき，

$$\begin{pmatrix} I_n & A_n \\ B_n & C_n \end{pmatrix} \boldsymbol{x}$$

の計算量を求めよ．

(6) ある正整数 p に対して $n = 2^p$ となるとする．問い (4) の式を再帰的に用い，n 次ベクトル $\boldsymbol{z}$ に対して $F_n \boldsymbol{z}$ を $O(n \log n)$ で計算するアルゴリズムを示せ．

問題 4

単一 CPU 計算機でページサイズが 4 kbyte のデマンドページングシステムを考える．

(1) 2 つのコード C1，C2 が配列 a を以下のようにアクセスしている．int 型のサイズは 4 byte とし，配列 a は，仮想メモリアドレス 0 番地から a[0][0], a[0][1], a[0][2], ... の順番で配置される．

```
int a[4][1024];
```

```
コード C1
register int   i, j;
for (i = 0; i < 4; i++) {
 for (j = 0; j < 4; j++) {
  a[i][j] = i*j;
 }
}
```

```
コード C2
register int   i, j;
for (j = 0; j < 4; j++) {
 for (i = 0; i < 4; i++) {
  a[i][j] = i*j;
 }
}
```

以下，4 つの仮想ページアクセスパターンを示す．整数値は仮想ページ番号を意味する．C1 および C2 のアクセスパターンはどれか？

A) 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
B) 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7
C) 0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3
D) 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3

(2) プロセス P_1, P_2, P_3 は仮想メモリ空間を共有し，仮想ページ番号 0 から 7 をアクセスする．プロセス実行前にこれら仮想メモリ空間は物理メモリに割り当てられていないとする．その他プロセス実行に必要な仮想メモリ空間は物理メモリに割り当てられ，これらはスワップアウトされない．共有メモリ領域に割り当て可能な空き物理メモリサイズは 16 kbyte (4 ページ分) とし，ページ置換アルゴリズム FIFO を用いてメモリ管理する．以下の問いに答えよ．

(a) 以下の実行順序で時刻 T_1 から時刻 T_7 の間，プロセスがスケジューリングされた時のページフォールトの回数を求めよ．例えば，時刻 T_1 から時刻 T_2 (T_1〜T_2) では，プロセス P_1 が仮想ページ番号 0, 1, 0, 1 の順にアクセスしている．

$T_1 \sim T_2$	$T_2 \sim T_3$	$T_3 \sim T_4$	$T_4 \sim T_5$	$T_5 \sim T_6$	$T_6 \sim T_7$
$P_1\{0, 1, 0, 1\}$	$P_2\{3, 2, 1, 0\}$	$P_3\{0, 3, 2, 3\}$	$P_1\{4, 5, 6, 7\}$	$P_2\{3, 2, 1, 0\}$	$P_3\{0, 1, 2, 3\}$

(b) 以下のように問い (a) とは異なる実行順序でプロセスがスケジューリングされた時のページフォールトの回数を求めよ．

$T_1 \sim T_2$	$T_2 \sim T_3$	$T_3 \sim T_4$	$T_4 \sim T_5$	$T_5 \sim T_6$	$T_6 \sim T_7$
$P_2\{3, 2, 1, 0\}$	$P_3\{0, 3, 2, 3\}$	$P_1\{0, 1, 0, 1\}$	$P_2\{3, 2, 1, 0\}$	$P_3\{0, 1, 2, 3\}$	$P_1\{4, 5, 6, 7\}$

(c) ワーキングセット WS(i, j) は時刻 T_i から時刻 T_j の間にアクセスされるページ集合を表現し，ワーキングセットサイズ WSS(i, j) は時刻 T_i から時刻 T_j の間のワーキングセットのサイズとする．例えば，問い (b) において時刻 T_3 から時刻 T_5 のワーキングセットおよびワーキングセットサイズは，$\mathrm{WS}(3, 5) = \{0, 1, 2, 3\}$，$\mathrm{WSS}(3, 5) = 4$ である．ワーキングセット WS(i, j) およびワーキングセットサイズ WSS(i, j) を用いて，スケジューリングによってページフォールトの回数に違いが生じる理由を説明せよ．

問題 2

ラベルの集合 L について，辺にラベルの付いた有限有向グラフ $G = (V, E)$ を考える．ただし，

- V は頂点の集合,
- $E \subseteq V \times L \times V$ はラベルの付いた辺の集合

である．また，L, V, E はすべて有限集合とし，$V \neq \emptyset$ かつ $L \neq \emptyset$，また $\mathcal{P}(L)$ は L の冪集合を表すものとする．

ここで， $Q, Q' \subseteq V \times V \times \mathcal{P}(L)$ を変数とし，次の手続き $(*)$ を考える．

$$
\begin{array}{l}
Q := \{\, (v, v, \emptyset) \mid v \in V \,\}; \\
\texttt{repeat} \quad [\\
\quad Q' := \{\, (v, w', A \cup \{a\}) \mid (v, w, A) \in Q \text{ and } (w, a, w') \in E \,\} \,\cup\, Q; \\
\quad \texttt{if} \quad Q = Q' \quad \texttt{then} \quad \texttt{done} \quad \texttt{else} \quad Q := Q' \\
]
\end{array}
$$

ここで $\texttt{repeat}[c]$ は，$\texttt{done}$ が実行されるまでの間，その内容 c を繰り返し実行することを表す．手続き $(*)$ を実行した後の，Q の値を Q_∞ と書く．

以下の問いに答えよ．

(1) 手続き $(*)$ が停止することを証明せよ．

(2) 2 つの頂点 $v, w \in V$ に対して，v から w に至る経路

$$v = v_0 \xrightarrow{a_0} v_1 \xrightarrow{a_1} \dots \xrightarrow{a_{n-1}} v_n = w$$

が存在するかどうか調べたい (ラベルは問わない)． このことは Q_∞ を用いてどのように判定できるか．

(3) ラベル $a \in L$ と頂点 $v \in V$ について，

v から始まる無限経路

$$v = v_0 \xrightarrow{a_0} v_1 \xrightarrow{a_1} \cdots$$

で，ラベル a の辺を通らないものが存在する

という性質が成立するかどうか調べたい．Q_∞ を用いて判定する方法を述べよ．

(4) ラベル $a \in L$ と頂点 $v \in V$ について，

v から始まる任意の無限経路

$$v = v_0 \xrightarrow{a_0} v_1 \xrightarrow{a_1} \cdots$$

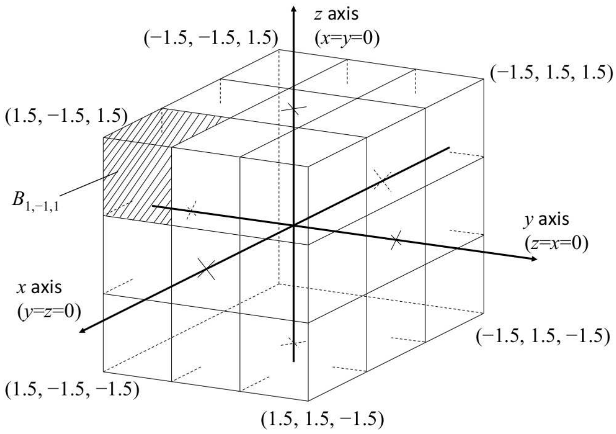
は次の条件をみたす： 任意の $n \geq 0$ に対して $m \geq n$ なる m が存在して $a_m = a$

という性質が成立するかどうか調べたい．Q_∞ を用いて判定する方法を述べよ．

(5) Q_∞ を並列計算する方法について，アイデアを述べよ．

問題 3

以下では，ルービック・キューブの xyz 空間内でのモデリングを考える．27 個の 1 辺の長さ 1 の立方体 $B_{i,j,k}$ $(i,j,k \in \{-1,0,1\})$ からなる立方体集合 A 中の各立方体を，ある初期配置から再配置していくことを考えよう．



初期配置において，各立方体 $B_{i,j,k}$ の中心は座標 (i,j,k) に置かれ，その各面は xy 平面，yz 平面，zx 平面のいずれかに必ず平行になるように配置されている．たとえば，立方体 $B_{1,-1,1}$ の初期状態での位置は，図に示すとおりである．なお，この初期状態で $B_{i,j,k}$ が配置されている立方体の位置を $L_{i,j,k}$ と表すものとする．

また，平面 $P_1, P_2, \ldots, P_6$ を それぞれ平面 $x = 1.5$，$x = -1.5$，$y = 1.5$，$y = -1.5$，$z = 1.5$，$z = -1.5$ と定める．各立方体の面のうち，初期配置において平面 P_i に載っている面を色 C_i $(i \in \{1,2,\ldots,6\})$ で塗り，これらの 6 つのいずれの平面にも載っていない面は色 C_0 で塗るものとする．なお，これらの色 C_i $(i \in \{0,1,\ldots,6\})$ はすべて相異なるものとする．

また，X_ℓ を，$L_{i,j,k}$ $(i=\ell)$ に配置されている 9 個の立方体の集合，Y_ℓ を，$L_{i,j,k}$ $(j=\ell)$ に配置されている 9 個の立方体の集合，Z_ℓ を，$L_{i,j,k}$ $(k=\ell)$ に配置されている 9 個の立方体の集合，と定める．ここで，立方体集合 A に対して，以下の $3 \times 3 = 9$ 通りの操作を行うことを考える．

(a_ℓ) X_ℓ に属する 9 つの立方体すべてを x 軸 $(y=z=0)$ に関して時計回りに 90 度回転させる．
(b_ℓ) Y_ℓ に属する 9 つの立方体すべてを y 軸 $(z=x=0)$ に関して時計回りに 90 度回転させる．
(c_ℓ) Z_ℓ に属する 9 つの立方体すべてを z 軸 $(x=y=0)$ に関して時計回りに 90 度回転させる．

たとえば，操作 (b_{-1}) によって，$L_{1,-1,1}$ に存在した立方体は $L_{-1,-1,1}$ に移動する．その際，元々平面 P_5 に載っていたその立方体の面は P_2 上に移動する．

また，各立方体 $B_{i,j,k}$ の位置および，それぞれの立方体の面の色の向きの組み合わせを A の状態とよぶ．すなわち，A 中のある立方体が回転しても，それらのすべての向きの面の色が同じであれば，二つの状態は区別しない．たとえば，$B_{0,0,0}$ はすべての面が色 C_0 で塗られており，また位置も不変であるため，どのような操作をしても操作前と操作後の $B_{0,0,0}$ の状態を区別することはできない．

このとき，以下の問いに答えよ．

(1) 初期状態から 2 つの操作（同じでもよい）を行って作られる A の状態は何通りあるか？

(2) 27個の立方体 $B_{i,j,k}$ $(i,j,k \in \{-1,0,1\})$ を，色 C_0 が塗られた面が 6 つの平面 P_i $(i \in \{1,2,\ldots,6\})$ に載らないようにして位置 $L_{i,j,k}$ $(i,j,k \in \{-1,0,1\})$ に再配置することを考える．このとき，操作 (a_ℓ), (b_ℓ), (c_ℓ) にとらわれることなく，各立方体を自由に移動・回転させて良い．また，異なる立方体は異なる位置に配置するものとする．このようにして得られる A の状態は何通りあるか．なお，$a!$，$a^b(a,b$ は整数$)$, およびかけ算の記法 $(\times)$ は解答に使ってよい (例：$5! \times 3^4 \times 7$)．

(3) (2) で求めた状態数のうち，$1/(2^3 \times 3^2 \times 5)$ の数の状態は，初期状態から操作 (a_ℓ), (b_ℓ), (c_ℓ) を繰り返すことで到達可能であることが知られている．このことを利用し，初期状態から到達可能だが，初期状態からその状態に到達するには最低 22 回以上の操作が必要な A の状態が必ず存在することを証明せよ．ただし，以下の事実を用いてよい．

$$1.58 < \log_2 3 < 1.59 \quad 2.32 < \log_2 5 < 2.33 \quad 2.80 < \log_2 7 < 2.81$$
$$3.45 < \log_2 11 < 3.46 \quad 3.70 < \log_2 13 < 3.71 \quad 4.08 < \log_2 17 < 4.09$$

問題 4

動的スケジューリングを用いる out-of-order instruction issue, out-of-order instruction completion プロセッサアーキテクチャについて，以下の問いに答えよ．

(1) パイプラインアーキテクチャを採用しても，一般にパイプライン段数に比例する性能向上が得られないのはなぜか．2~3 行で説明せよ．

(2) 動的スケジューリングの利点と欠点を，静的スケジューリングと対比して 3~5 行で述べよ．

(3) 命令キューは，発行（issued）されたが完了（completed）していない命令を保持するバッファである．　命令が命令キューに挿入される条件と，命令が命令キューから削除される条件を述べよ．

(4) 例を用い，実行中に命令がどのようにスケジュールされるか説明せよ．

(5) 動的スケジューリングを用いるプロセッサの性能を向上させるためには高精度の分岐予測が重要である．この理由を述べよ．

問題 3

x 座標が相異なる n 個の点からなる x-y 平面上の点集合 S を考える．$\mathrm{dist}(v,w)$ は 2 点 v, w 間のユークリッド距離を表すものとする．このとき，以下の問いに答えよ．ただし丸め誤差は考えないものとする．

(1) S の n 個の点を x 座標について昇順にソートするアルゴリズムを 1 つ示し，その時間計算量を述べよ．

なお，各点は x 座標を表す浮動小数点数と y 座標を表す浮動小数点数からなる構造体として表現され，点集合 S はその構造体の長さ n の配列として与えられているものとする．

(2) S 中の相異なる 2 点からなるすべての組 $\langle v,w\rangle$ $(v \neq w)$ のうち，$\mathrm{dist}(v,w)$ が最小である組の距離の値を ℓ_S とする．このとき，平面上のいかなる点 (x^*,y^*) に対しても，$x^* \leq x \leq x^* + \ell_S/2$, $y^* \leq y \leq y^* + \ell_S/2$ で表される正方形内に S の点は高々1 つしかないことを示せ．

(3) S の n 個の点を x 座標について昇順にソートしたものを v_1, v_2, …, v_n とする．このとき，

$$P = \{v_1, v_2, \ldots, v_{\lfloor n/2 \rfloor}\}\ , \qquad Q = \{v_{\lfloor n/2 \rfloor+1}, v_{\lfloor n/2 \rfloor+2}, \ldots, v_n\}$$

とする．ただしここで $\lfloor x \rfloor$ は $x-1 < \lfloor x \rfloor \leq x$ なる唯一の整数を表す．

P 中の相異なる 2 点からなる組 $\langle s,t\rangle$ $(s \neq t)$ のうち，$\mathrm{dist}(s,t)$ が最小である組の距離の値を ℓ_P とする．点 $q \in Q$ を一つ固定したとき，$\mathrm{dist}(p,q) < \ell_P$ を満たすような点 $p \in P$ の個数を考えると，この個数は 8 個を超えないことが問い (2) を用いてわかる．このことを証明せよ．

(4) P の点 $p \in P$ の中で，点 $q \in Q$ からの距離 $\mathrm{dist}(p,q)$ が最小となるもの（複数ある場合はそのうちの 1 つ）を $\mathrm{nearest}_P(q)$ とおく．

ℓ_P の値が与えられているものとするとき，

$$\mathrm{dist}(\mathrm{nearest}_P(q), q) < \ell_P$$

を満たすようなすべての $q \in Q$ の集合を $\{q_1, q_2, \ldots, q_m\}$ とする．これらの点に対して

$$\mathrm{nearest}_P(q_1),\ \mathrm{nearest}_P(q_2),\ \ldots,\ \mathrm{nearest}_P(q_m)$$

をすべて求めたい．そのための効率のよいアルゴリズムとその時間計算量を示せ．

(5) S に対し ℓ_S を求めるアルゴリズムとその時間計算量を示せ．

問題 4

以下にスタック計算機シミュレータの一部を示す.

```
typedef unsigned char ub;
typedef signed char sb;
sb sim(ub *code, ub *dp, ub *sp)
{
 sb  disp; sb  run = 1, tmp;
 while (run) {
   switch(*code++) {
   case 00: /* push data */
     disp = *code++; *--sp = *(dp + disp);
     break;
   case 01: /* dup */
     --sp; *sp = *(sp + 1);
     break;
   case 02: /* swap */
     tmp = *sp;
     *sp = *(sp + 1); *(sp + 1) = tmp;
     break;
   case 03: /* add */
     tmp = (sb)*sp + (sb) *(sp + 1);
     *++sp = tmp;
     break;
```

```
   case 04: /* ... */
     tmp = *sp == *(sp + 1);
     *++sp = tmp;
     break;
   case 05: /* ... */
     tmp = *sp; disp = *code++;
     if (tmp == 0) code += disp;
     break;
   case 06: /* ... */
     disp = *code++; code += disp;
     break;
   case 07: /* stop */
     run = 0; break;
   case 255:
     abort(); /* never returns */
     break;
   }
 }
 return (sb) *sp;
}
```

以下の問いに答えよ.

(1) メモリアドレス 0 番地から以下のような機械語コード及びデータが格納されている. 各要素は 1 byte である.

```
 0 番地: 00, 01, 01, 03, 07, 00, 00, 00, 00, 00,
10 番地: 00, 00, 00, 00, 00, 00, 00, 00, 00, 00,
20 番地: 00, 10, -1, 00, 00, 00, 00, 00, 00, 00
```

以下のように実行した時の返り値を求めよ.

```
sim((ub*) 0, (ub*) 20, (ub*) 20)
```

(2) アセンブリ言語を設計し, 以下 (†) の機械語コード及びデータをアセンブリ言語で記述せよ.

(†)
```
 0 番地: 00, 01, 00, 01, 00, 02, 03, 05, 07, 02,
10 番地: 00, 03, 03, 02, 06, -12, 02, 07, 00, 00,
20 番地: 00, 10, -1, 01, 00, 00, 00, 00, 00, 00
```

(3) 上記 (†) を以下 (‡) のように実行するとアボートする. 理由を述べよ.

(‡)
```
sim((ub*) 0, (ub*) 20, (ub*) 20)
```

(4) 上記 (†) をアボートを避けながら期待通りに動作させるためには, (†) または (‡) をどう修正すれば良いか述べよ.

(5) 問い (3) のような問題が発生しないようにするために, スタック計算機シミュレータをどのように改良すれば良いか議論せよ.

問題 1

Alice と Bob の二人のプレイヤーの間で行われるゲームを考える．このゲームでは，それぞれのプレイヤーは2つの手 s_1, s_2 のうちひとつを選ぶ．ただし，お互いに相手がどの手を選ぶかは事前にはわからないものとする．

いま，Alice と Bob が選んだ手をそれぞれ S_A, S_B と表し，$S_A = s_i$，$S_B = s_j$ の際に Alice, Bob が獲得する得点をそれぞれ $f_A(s_i, s_j)$，$f_B(s_i, s_j)$ と表す．

以下の問いに答えよ．

(1) $f_A(S_A^*, S_B^*) \geq f_A(S_A, S_B^*)$ が任意の S_A に対して成り立ち，かつ，$f_B(S_A^*, S_B^*) \geq f_B(S_A^*, S_B)$ が任意の S_B に対して成り立つ $S^* = (S_A^*, S_B^*)$ を均衡点とよぶ．

いま，$f_A(s_i, s_j)$ を (i, j) 成分に持つ行列を M_A （すなわち $M_A = \begin{pmatrix} f_A(s_1,s_1) & f_A(s_1,s_2) \\ f_A(s_2,s_1) & f_A(s_2,s_2) \end{pmatrix}$），また $f_B(s_i, s_j)$ を (i, j) 成分に持つ行列を M_B とする．次の (a)，(b) それぞれにおいて，均衡点があればそれらをすべて与えよ．なければその理由を示せ．

(a) $M_A = \begin{pmatrix} 5 & -4 \\ 6 & -3 \end{pmatrix}$ ， $M_B = \begin{pmatrix} 5 & 6 \\ -4 & -3 \end{pmatrix}$

(b) $M_A = \begin{pmatrix} -1 & 4 \\ 0 & 2 \end{pmatrix}$ ， $M_B = \begin{pmatrix} -1 & 0 \\ 4 & 2 \end{pmatrix}$

(2) $M_A = \begin{pmatrix} 3 & -2 \\ 2 & 1 \end{pmatrix}$ ， $M_B = \begin{pmatrix} 2 & -1 \\ 0 & 3 \end{pmatrix}$ で定義されるゲームにおいて，Alice と Bob のそれぞれは，手 s_1 を確率 p, q （ただし $0 \leq p < 1/2$，$0 \leq q \leq 1$）で選択するものとする．

(a) Alice と Bob が獲得する得点の期待値をそれぞれ求めよ．

(b) Bob の獲得する得点の期待値を最大にする q の値を求めよ．

次に，得点が問い (1-a) で定義されるゲームを無限回繰り返す．この際，n 回目のゲームで獲得する得点はもともとの得点の d^{n-1} 倍であるとする（ただし $0 < d < 1$）．プレイヤーは次の 2 つの戦略のうちいずれかをとるものとする．

- （戦略 t_1） 常に s_1 を選ぶ．
- （戦略 t_2） 最初は s_1 を選び，相手が s_1 を選ぶ限り s_1 を選ぶ．相手が s_2 を選んだ次のゲームでは s_2 を選ぶ．一度でも自分で s_2 を選んだら，その後は s_2 を選び続ける．

以下の問いに答えよ．

(3) Alice, Bob が共に戦略 t_1 をとった場合の Bob の総得点 $\tilde{f}_B(t_1, t_1)$ を求めよ．

(4) Alice, Bob が共に戦略 t_2 をとったが，k 回目のゲームで Bob が誤って手 s_2 を選んだ場合を考える．この際の Alice, Bob の総得点をそれぞれ $\tilde{f}_A^*(t_2, t_2)$，$\tilde{f}_B^*(t_2, t_2)$ と表す．

(a) $\tilde{f}_A^*(t_2, t_2)$ を $\tilde{f}_B^*(t_2, t_2)$ と k を用いて表せ．

(b) $\tilde{f}_B^*(t_2, t_2) > \tilde{f}_B(t_1, t_1)$ が成り立つために d がみたすべき条件を求めよ．

問題 3

それぞれの棒に重さと長さがある n 個の棒の集合 $B = \{b_1, b_2, \ldots, b_n\}$ を考える．棒 b_i それぞれの重さを w_i，長さを ℓ_i とする．このとき，長さの和を L 以下におさえたまま，B から何本かの棒を選んだ時に，ありえる総重量の最大値を $W_L(B)$ とし，これを求める問題を考える．

以下の問いに答えよ．

(1) 重さがそれぞれ $2, 3, 6, 5, 7$，長さが同じ順にそれぞれ $1, 2, 3, 4, 5$ である棒の集合 B に対して，$W_5(B)$ を求めよ．

(2) すべての長さ ℓ_i および L が正の整数である場合を考える．また，$1 \leq k \leq n$ なる k のそれぞれに対して $B_k = \{b_1, b_2, \ldots, b_k\}$ とおく．何らかの方法で計算した $L+1$ 個の値 $W_j(B_{k-1})$ $(0 \leq j \leq L)$ がすでに得られていると仮定する．このとき，$0 \leq r \leq L$ なる r のそれぞれに対し $W_r(B_k)$ を計算する方法を述べよ．

(3) すべての長さ ℓ_i が正の整数であるとしたときに，整数 L に対して $W_L(B)$ を計算するアルゴリズムを述べ，その時間計算量を n, L を用いて表せ．

(4) すべての長さ ℓ_i が，適当な正の整数の組 c_i, d_i を用いて $\ell_i = c_i + d_i\sqrt{2}$ と表されるとしたときに，整数 L に対して $W_L(B)$ を計算するアルゴリズムを述べ，その時間計算量を n, L を用いて表せ．

問題 3

辺にコストが設定されている連結な無向有限グラフ $G = (V, E)$ が与えられたときに，すべての頂点を連結し，かつ閉路を含まないような部分グラフでコスト最小になるものを以下のようなアルゴリズムによって求める問題を考える．

- 1つの頂点 $s \in V$ を任意に選んで，集合 U を $U = \{s\}$ とする．
- 以下の性質 $(*)$ をみたすテーブルを用意する．

  $(*)$ テーブルは $|V \setminus U|$ 個のエントリからなる．各エントリは，頂点 $v \in V \setminus U$，集合 U 中の頂点の中で頂点 v に最も近いもの u （u を v の最寄頂点とよぶ），および v, u 間の距離 $d(v, u)$ の3つ組 $\left(v,\ u,\ d(v, u)\right)$ からなる．

  ここで，頂点の間の距離とは，その頂点間を結ぶ辺のコストのことを指すものとする．直接辺で結ばれていない場合の距離は ∞ とする．

  初期状態においては，各頂点 $v \in V \setminus U$ に対して，エントリ $\left(v,\ s,\ d(v, s)\right)$ をテーブルに格納する．

- 空の辺の集合 T を用意し，以下の手順を $V \setminus U$ が空になるまで繰り返し，最後に得られた T が求める部分グラフを構成する．
  - 集合 $V \setminus U$ の頂点のうち，テーブルに格納されている距離のもっとも小さい頂点 v を選び，v とその最寄頂点 $u \in U$ を結ぶ辺を T に加える．またこのとき，$\boxed{\text{(a)}}$ を $\boxed{\text{(b)}}$ に加え，さらにそれに合わせてテーブルの内容を性質 $(*)$ をみたすように更新する．

(1) アルゴリズムが正しく動作するように，空欄 $\boxed{\text{(a)}}$ および $\boxed{\text{(b)}}$ をうめよ．

(2) 上記のアルゴリズムによって，グラフの辺のコストがすべて非負であった場合に，求めたい部分グラフが得られることを説明せよ．

(3) 上記のアルゴリズムは，辺のコストに負のものがあっても正しく動作するか，答えよ．正しく動作しない場合には反例を示せ．正しく動作する場合には，根拠を示せ．

(4) グラフが密な場合 ($|E| \approx |V|^2$) には，どのようなデータ構造を用いるべきか，説明せよ．またその場合の時間計算量とその根拠を示せ．

(5) グラフが疎な場合 ($|E| = O(|V|)$) には，どのようなデータ構造を用いるべきか，説明せよ．またその場合の時間計算量とその根拠を示せ．

問題 4

3進数の演算を行う論理回路を考える．ここでは，3進数の各桁は0, 1, 2の値のいずれかであり，3進数の1桁を2ビットで表現するものとする．

以下の問いに答えよ．

(1) 10進数の20および65を3進数で表せ．

(2) 3進数1桁の半加算器および全加算器をAND，OR，NOTゲートを用いて設計せよ．まず真理値表を示し，次に論理回路を示せ．

(3) 3進数4桁の加算器を，問い(2)で設計した半加算器および全加算器を用いて設計せよ．

(4) 3進数16桁のキャリールックアヘッド加算器を設計せよ．

問題 4

P 台のプロセッサがあり，0 から $P-1$ までの番号がついている．プロセッサ i は，値が格納されている長さ N の整数配列 a_i を持っているとする．これらのプロセッサが互いに通信しながら，それぞれの持つ配列 $a_0, a_1, \ldots, a_{P-1}$ の要素ごとの総和を求め，プロセッサ 0 の配列 a_0 に格納することを考える．つまり，計算前のプロセッサ i の配列を a'_i，計算後のそれを a''_i と書くと，

$$a''_0[j] = \sum_{i=0}^{P-1} a'_i[j] \ , \qquad j = 0, 1, \ldots, N-1$$

となるようにしたい．以下，これを総和計算とよぶ．

ネットワークに対しては次のような仮定をおく：任意のペア (i_s, i_d) に対し，プロセッサ i_s からプロセッサ i_d に長さ L の整数配列を配送するために $\alpha + \beta L$ の時間がかかる．これは仮にプロセッサ i'_s から i'_d への転送が同時に行われていても，$\{i_s, i_d\} \cap \{i'_s, i'_d\} = \emptyset$ である限り同じ時間がかかるものとする．また，送受信プロセッサが重なる複数の送受信は同時にはできない（早く始まった通信が終了するまで待たされる）ものとする．さらに，整数の和の計算をするのは通信に比べて十分早いのでその所要時間は無視できるものとする．

所要時間は，最初に行われる通信の開始時刻から，プロセッサ 0 上での値がすべて求まるまでの時間とする．

以下の問いに答えよ．

(1) 次のようなアルゴリズムで総和計算をした場合の所要時間を求めよ．

　　$i = 1, 2, \ldots, P-1$ に対し逐次に以下の処理を行う {
　　　　プロセッサ i がプロセッサ 0 に a_i 全体を送信する．
　　　　プロセッサ 0 に a_i が届いたら，a_i と a_0 の要素ごとの和を a_0 に格納する．
　　}

(2) P は 2 のべきとし，長さ N は十分短いとする．総和計算を所要時間 $O(\log P)$ で実現するアルゴリズムを考えよ．アルゴリズムを簡潔に説明し，所要時間を示せ．問い (1) で示したアルゴリズムと性能を比べよ．

(3) 次に長さ N は十分大きいものとして，次のアルゴリズムを考える．ここで K はすべてのプロセッサに共通の定数であり，各プロセッサは配列を K 個の部分配列に等分する．

- プロセッサ $P-1$ は，K 個の部分配列を順次プロセッサ $P-2$ に送る．
- プロセッサ i（ただし $1 \le i \le P-2$）は，

　　プロセッサ $i+1$ から部分配列が届いたら，対応する自分の部分配列と要素ごとの和をとったうえ，その結果をプロセッサ $i-1$ に送る

　ことを K 回繰り返す．
- プロセッサ 0 は，

　　プロセッサ 1 から部分配列が届いたら，対応する自分の部分配列と要素ごとの和をとったうえ，その結果を a_0 に格納する

　ことを K 回繰り返す．

このアルゴリズムの所要時間を示し，このアルゴリズムを最適にする K の値を見積もれ．整数値である必要はない．

(4) 問い (3) のアルゴリズムが，問い (2) で答えたアルゴリズムよりも高速な場合があるかどうか検討せよ．