

問 1 DNS(ドメインネームシステム)に関する次の記述を読んで、設問 1～3 に答えよ。

インターネットで使われるドメイン名及び IP アドレスは、DNS を利用して管理されている。DNS は、多数の DNS サーバで構成される  データベースであり、  
5 ルート DNS サーバを頂点とし、ドメイン名空間と呼ばれるツリー構造を構成している。インターネットでは、 と  を考慮して、13 のルート DNS サーバが配置されている。

DNS のツリー構造の最上位に位置するルート DNS サーバの配下には、ドメイン名  
(例:jp ドメイン, co.jp ドメインなど)に対応した DNS サーバがある。あるドメイン  
10 名を管理する DNS サーバに関する情報は、ツリー構造の  のドメイン名を管理する DNS サーバが保持している。ホストに関する情報は、そのホストが所属するドメイン名を管理する DNS サーバが保持している。

ドメイン名の中で、www.example.co.jp のように特定のホストを表現したドメイン名を  と呼ぶ。

15 DNS の最も一般的な使われ方は、 から IP アドレスへの変換である。これを“ホストの名前解決”と呼ぶ。

一つのドメインを管理する DNS サーバは、通常は  を考慮して 2 台のサーバで構成される。一方を  , もう一方を  と呼ぶ。

20 ホストマシンをインターネットに接続する方法には、大きく分けて“直接接続方式”と“間接接続方式”がある。直接接続方式では、グローバル IP アドレスを使用して、インターネットに直接接続する。間接接続方式では、プライベート IP アドレスを使用し、ルータの NAT 機能でグローバル IP アドレスに変換したり、プロキシサーバで中継したりして、インターネットに間接的に接続する。

25 設問 1 本文中の   に入れる適切な字句を解答群の中から選び、記号で答えよ。

解答群

- |      |          |   |           |   |         |
|------|----------|---|-----------|---|---------|
| ア    | ARPA     | イ | FQDN      | ウ | IANA    |
| エ    | オブジェクト指向 | オ | 下位階層      | カ | 可用性     |
| 30 キ | キャッシュサーバ | ク | サブドメイン名   | ケ | 集中型     |
| コ    | 上位階層     | サ | セカンダリサーバ  | シ | セキュリティ  |
| ス    | 帯域制御     | セ | バックアップサーバ | ソ | 負荷分散    |
| タ    | プライマリサーバ | チ | 分散型       | ツ | リレーショナル |

35 設問 2 クライアント PC (以下、PC という) を直接接続方式でインターネットに接続する場合の、ホストの名前解決の動作として正しいものを解答群の中からすべて選び、記号で答えよ。ここで、PC には、ホストの名前解決のために DNS サーバ X (以下、X という) が指定されているものとする。

40 解答群

- ア PC には, X が IP アドレスを返す。
- イ PC には, 要求されたホストの情報を保持する DNS サーバが IP アドレスを返す。
- ウ PC には, ルート DNS サーバが IP アドレスを返す。
- エ X から問合せを受けたルート DNS サーバは, ドメイン名空間においてルート
- 5 DNS サーバの一つ下位にある DNS サーバに関する情報を X に返す。
- オ X から問合せを受けたルート DNS サーバは, 要求されたホストの情報を保持する DNS サーバに関する情報を X に返す。
- カ X が, 要求されたホスト及び IP アドレスの情報をキャッシュしていた場合には, X はその情報を直ちに PC に返す。
- 10 キ X は, 要求されたホストの情報を保持する DNS サーバから, ホストに対応する IP アドレスを得る。

設問 3 次の文章は, 社内 LAN 上の PC を間接接続方式でインターネットに接続する場合の, インターネット上の Web サーバへのアクセスと, SMTP を利用した電子メール送信の際の, ホストの名前解決の動作を記述したものである。 h

- 15 i に入れる適切な字句を解答群の中から選び, 記号で答えよ。  
なお, e には本文と同じ字句が入るものとする。

インターネット上の Web サーバへのアクセスのために, 社内 LAN とインターネットの両方にアクセス可能なプロキシサーバを設置する。Web ブラウザからこの

20 プロキシサーバにアクセスするように, PC の設定をする。プロキシサーバは, PC から要求された h に含まれる Web サーバの e で DNS を引き, その Web サーバの IP アドレスを得る。その後, プロキシサーバは得られた IP アドレスで, 目的とする Web サーバにアクセスし, Web サーバからの応答を PC に返す。

- 25 電子メールを送信するために, 社内 LAN とインターネットの両方にアクセス可能な SMTP サーバを設置する。PC は, この SMTP サーバに向けて電子メールを送信する。電子メールを受信した SMTP サーバは, 電子メールに書かれた送信先 i に含まれる e で DNS を引き, その電子メールの送信先となる SMTP サーバの IP アドレスを得る。その後, 送信元の SMTP サーバは, その
- 30 IP アドレスが示す送信先の SMTP サーバあてに, 電子メールを送信する。

解答群

- |           |              |             |
|-----------|--------------|-------------|
| ア IP アドレス | イ URL        | ウ グローバルアドレス |
| エ 経路情報    | オ プライベートアドレス | カ メールアドレス   |

問 2 ソフトウェアの構成管理に関する次の記述を読んで、設問 1～3 に答えよ。

B 社では、自社の基幹システムの再開発を行っている。現在、ユニットテストが完了し、結合テストを行っている段階である。結合テスト時のプロジェクト体制を図 1 に、プログラムのテスト及びバグ修正の手順を図 2 に示す。

プロジェクトでは、結合テスト時に発見したバグを修正するに当たって、プログラムの一貫性を保ち、かつ、管理の効率向上を図るために、構成管理ツールを導入している。導入した構成管理ツールは、表 1 に示す機能をもつ。

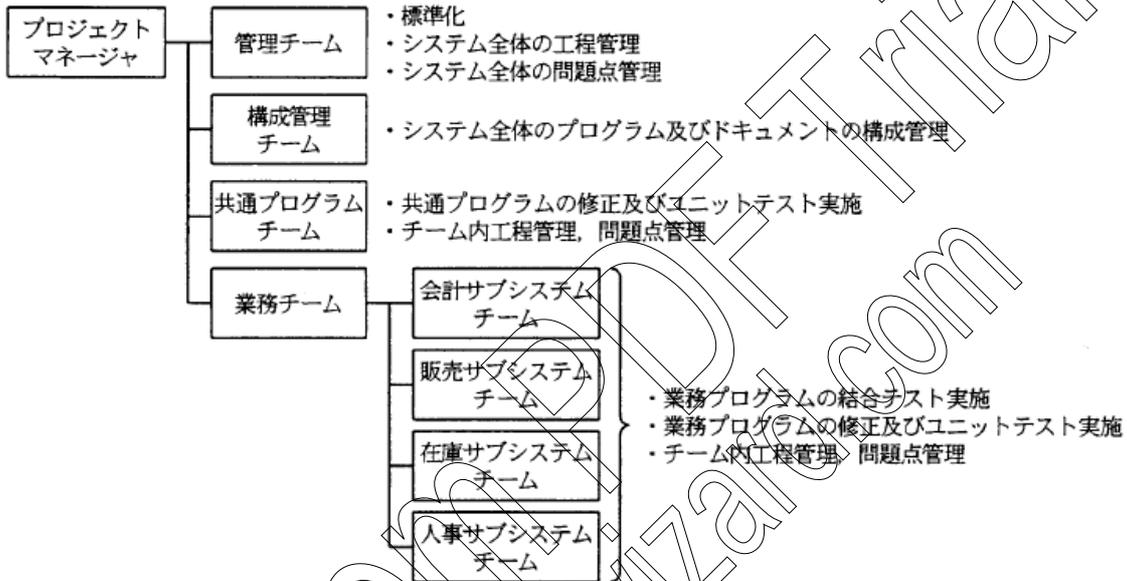


図 1 結合テスト時のプロジェクト体制

表 1 構成管理ツールの機能

項番	機能	機能の説明
1	最新版管理の機能	最後に更新されたソースプログラムを管理し、ユニットテスト完了、結合テスト完了などの完了状態を記録する。
2	履歴管理の機能	更新時のソースプログラムに更新番号を付与し、プログラムの更新者、更新年月日時分秒を履歴として管理する。
3	<input type="text" value="a"/> , <input type="text" value="b"/> の機能	<input type="text" value="a"/> を行った修正担当者に対しては、構成管理ツールで管理しているソースプログラムの修正を許可し、 <input type="text" value="b"/> を行うまではほかの担当者のプログラム修正を禁止する。
4	統計情報の取得・保存、 統計レポート作成機能	プログラム別の各種統計情報を取得・保存し、統計レポートの作成を行う。

10

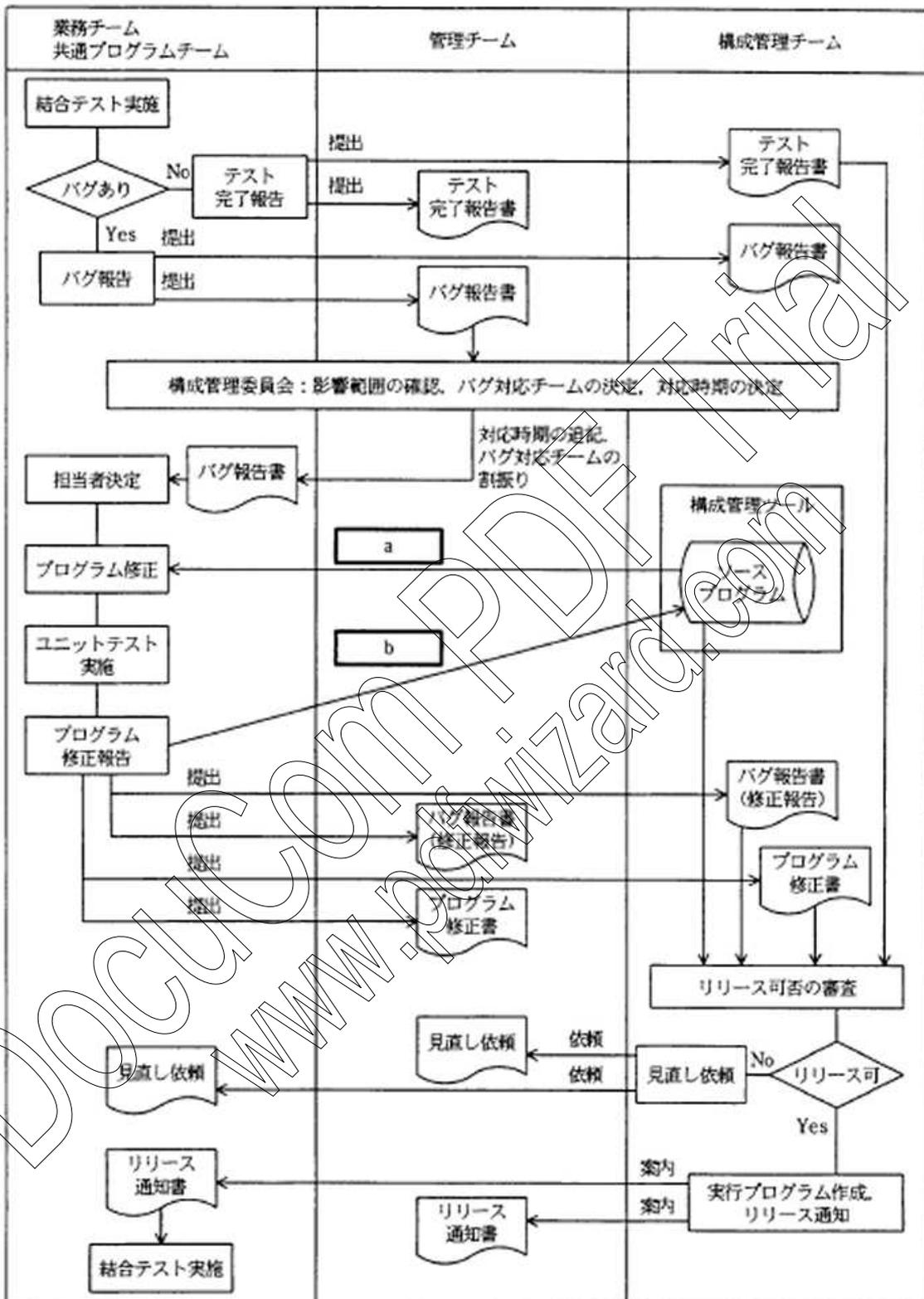


図 2 プログラムのテスト及びバグ修正の手順

設問 1 表 1 及び図 2 中の **a** , **b** に入れる適切な字句を, それぞれ 7 字以内のカタカナで答えよ。

設問2 C君は、共通プログラムチームのメンバである。C君は、修正を担当した共通プログラムの動作を確認するため、業務プログラムを用いてテストを実施したところ、業務プログラムのバグを発見した。このプロジェクト体制において、C君がとるべき行動として適切なものを解答群の中から一つ選び、記号で答えよ。

5 解答群

ア 業務プログラムのバグへの対応は特に何もせず、自分の担当している共通プログラムの修正、ユニットテストを完了させる。

イ 業務プログラムのバグ報告書を作成し、管理チームと構成管理チームに提出する。

10 ウ 担当している共通プログラムと併せて、業務プログラムを修正し、ユニットテストを完了させる。

エ 当該業務プログラムの担当者にバグの内容を口頭で伝え、バグ報告書を作成してもらう。

設問3 使用している構成管理ツールで取得できる統計レポートの内容及び担当者からの報告内容を用いて可能性が推定できる事象の内容を表2にまとめる。表2中の  ～  に入れる適切な字句を解答群の中から選び、記号で答えよ。

表2 取得できる統計情報と統計内容

入手元	記号	統計レポートの内容及び報告内容	統計レポートから可能性が推定できる事象	
			内容	併せて参照する内容
構成管理ツール	①	担当者別のプログラム本数	<input type="text" value="c"/>	⑧, ⑨
	②	担当者別の修正期間		
	③	プログラム別の最終更新年月日時分秒	<input type="text" value="d"/>	⑨, ⑩
	④	プログラム別の修正回数	<input type="text" value="e"/>	⑧, ⑩
	⑤	プログラム別の修正期間	<input type="text" value="f"/>	⑧, ⑨
	⑥	プログラム修正期間の平均・最大		
	⑦	プログラム別、修正回別の修正行数	<input type="text" value="e"/> <input type="text" value="g"/>	⑧, ⑨, ⑩
	⑧	プログラム別のステップ数	-	-
担当者からの報告	⑨	プログラム別のテスト開始・完了状況	-	-
	⑩	プログラム別のチェックリスト消化状況	-	-

解答群

- 20 ア 修正の難易度                      イ 担当者の負荷状況  
 ウ テストの進捗状況                エ バグの多いプログラム  
 オ プログラムの品質向上傾向



```

<!DOCTYPE 商品 [
  <!ELEMENT 商品 (商品コード, 商品名, 商品種別, 単価) >
  <!ELEMENT 商品コード (#PCDATA) >
  <!ELEMENT 商品名 (#PCDATA) >
  <!ELEMENT 商品種別 (#PCDATA) >
  <!ELEMENT 単価 (#PCDATA) >
] >
<商品>
  <商品コード= "A-123" >
  <商品名= "ノートパソコンI" >
  <商品種別= "Note PC" >
  <単価= "125000" >
</商品>

```

```

(3) <?xml version="1.01" ? >
<!DOCTYPE 商品 [
  <!ELEMENT 商品 (単価) >
  <!ATTLIST 商品 商品コード CDATA #REQUIRED
                商品名 CDATA #REQUIRED
                商品種別 CDATA #REQUIRED >
  <!ELEMENT 単価 (#PCDATA) >
] >
<商品
  商品コード= "A-123"
  商品名= "ノートパソコン"
  商品種別= "Note PC" >
  <単価>125000</単価>
</商品>

```

- 30 解答群  
 ア “妥当な文書”      イ “整形形式の文書” であるが “妥当な文書” ではない  
 ウ どちらでもない

設問3 図1に示す発注データを，コンピュータ間で受け渡す場合について考える。

発注者	ABC商事	発注年月日	2006-10-20
		希望納期	2006-10-31
注文明細			
商品コード	商品名	個数	
C-345	CPU	10	
M-666	メモリ	20	

図 1 発注データ

この発注データを XML 文書で表現したものを図 2 に示す。図 2 中の  に入れる適切な字句と、 ~  に入れる適切な XML 表記を答えよ。

```

<?xml version="1.0" ?>
<!DOCTYPE 注文 [
  <!ELEMENT 注文 (注文明細+)>
  <!ATTLIST 注文 発注者 CDATA #REQUIRED
    発注年月日 CDATA #REQUIRED
    希望納期 CDATA #REQUIRED >
  <!ELEMENT  (商品コード, 商品名, 個数)>
  <!ELEMENT 商品コード (#PCDATA)>
  <!ELEMENT 商品名 (#PCDATA)>
  <!ELEMENT 個数 (#PCDATA)>
] >


<商品コード>C-345</商品コード>
<商品名>CPU</商品名>
<個数>10</個数>


<商品コード>M-666</商品コード>
<商品名>メモリ</商品名>
<個数>20</個数>

</注文>
    
```

図 2 発注データの XML 文書

問 4 ディスク装置の構成技術である RAID に関する次の記述を読んで、設問 1, 2 に答えよ。

RAID とは、Redundant Array of Independent (又は、Inexpensive) Disks の略で、独立したディスク装置を複数台用いて、可用性が高く、高速で大容量の補助記憶装置

5 (以下、ディスクアレイという) を構築する技術である。RAID には、実現方法によってレベルがある。代表的なものを次に示す。

RAID0 : ストライピングとも呼ばれる。データを複数のディスク装置に分散して配置することで、読み込み/書き込み速度を向上させる。RAID0 には冗長性がないので、ディスクアレイを構成しているディスク装置のうち、いずれか 1 10 台にでも障害が発生すると、ディスクアレイは稼働不能となる。

RAID1 : ミラーリングとも呼ばれる。同じデータを複数のディスク装置に書き込むことで、いずれか 1 台のディスク装置に障害が発生しても、ディスクアレイとして稼働するようにしている。

RAID4 : RAID0 と同じように複数のディスク装置をストライピングで構成するが、15 パリティと呼ばれるエラー訂正情報を保持するパリティディスクがあり、いずれか 1 台のディスク装置に障害が発生してもディスクアレイとして稼働する。読み込みはストライピング効果で高速であるが、書き込みはパリティディスクにアクセスが集中するのであまり速くない。

RAID5 : RAID4 を改良し、固定であったパリティディスクを分散させることで、パリティディスクへのアクセスの集中を防ぎ、高速化を実現している。

RAID01 : ストライピングしたディスク装置群を一つの単位としてミラーリングすることで、RAID0 の高速性を保ちながら、高可用性を実現している。

RAID10 : ミラーリングしたディスク装置群を一つの単位としてストライピングすることで、RAID1 の高可用性を保ちながら、高速性を実現している。

25 設問 1 図は、ディスクアレイの構成を RAID のレベルごとに表したものである。①～⑤の RAID レベルを、本文で示した六つの RAID レベルの中から選択し、それぞれ答えよ。

なお、図中の数字はデータのブロック番号、P はパリティを表している。

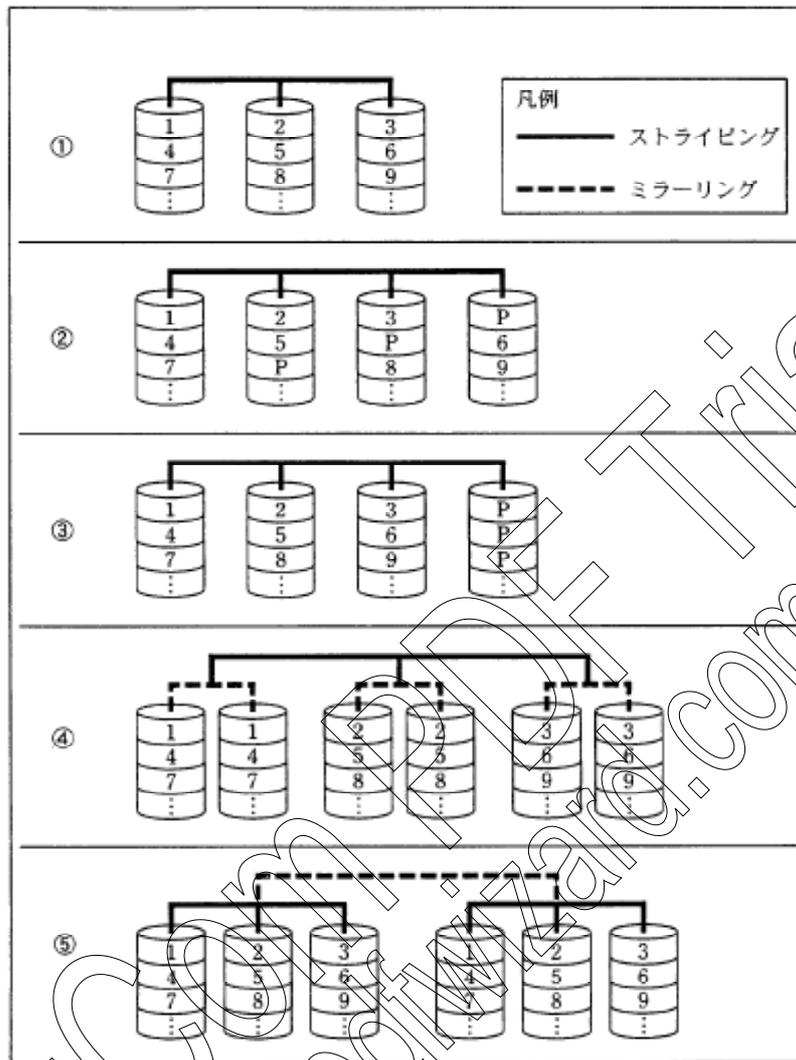


図 データレイの構成

設問 2 データレイの可用性について考える。2 台のディスク装置で RAID0 を構成した場合は、2 台のディスク装置のうち、どちらかに障害が発生するとデータレイは稼働不能となる。したがって、ディスク装置 1 台の稼働率を  $u$  とすると、データレイの稼働率は  $u^2$  となる。一方、2 台のディスク装置で RAID1 を構成した場合は、どちらかのディスク装置が稼働していればデータレイとして稼働する。したがって、データレイの稼働率は、2 台のディスク装置がともに稼働している場合 ( $u^2$ ) と、どちらか 1 台だけが稼働している場合 ( $2u(1 - u)$ ) の和  $u^2 + u(1 - u)$  となる。

RAID01, RAID10 については、まず、単位となるディスク装置群の稼働率を求め、次に、その値に基づいて全体のデータレイの稼働率を求める。

表は、1 台当たりの容量が 100 G バイトのディスク装置を用いて、実効容量 400 G バイトのデータレイを構成する場合の、最小構成時のディスク装置台数、データレイが稼働不能となる最小障害ディスク装置台数、及びデータレイ

イの稼働率を, RAID レベルごとにまとめたものである。

表中の  ,  ,  ,  に入れる適切な数値を答えよ。また,  ,  に入れる適切な式を解答群の中から選び, 記号で答えよ。ここで, ディスク装置 1 台の稼働率はすべて  $u$  とする。

表 RAID レベルごとのディスクアレイの構成と可用性

RAID レベル	最小構成時の ディスク装置台数	ディスクアレイが稼働 不能となる最小障害 ディスク装置台数	ディスクアレイの 稼働率
RAID0	4	1	$u^4$
RAID4, RAID5	<input type="text" value="a"/>	2	<input type="text" value="b"/>
RAID01	<input type="text" value="c"/>	2	<input type="text" value="d"/>
RAID10	<input type="text" value="e"/>	<input type="text" value="f"/>	$(u^2 + 2u(1-u))^4$

5

b, d に関する解答群

ア  $u^3 + 3u^2(1-u)$

イ  $u^4 + 4u^3(1-u)$

ク  $u^5 + 4u^4(1-u)$

エ  $u^5 + 5u^4(1-u)$

オ  $(u^2 + 2u(1-u))^4$

カ  $u^3 + 2u^4(1-u^4)$

10

問 5 2分探索木に関する次の記述を読んで、設問 1～4に答えよ。

ノードが一つずつデータをもつ2分探索木は、どのノードNから見ても、左側の部分木のノードがもつデータはすべてNのデータよりも小さく、逆に右側の部分木のノードがもつデータはすべてNのデータよりも大きい、という性質をもつ。ここでは、ノードがもつデータはすべて自然数で重複がないものとする。図1に2分探索木の例を示す。ノード内の数が、データを表している。

2分探索木に対して、表1に示す二つの操作を定義する。

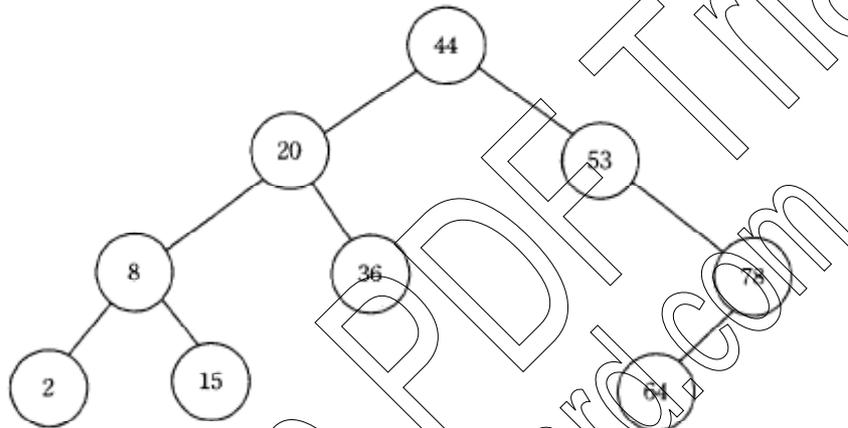


図 1 2分探索木の例

表 1 2分探索木に対する操作

操作名	操作内容
lookup	指定したデータをもつノードが2分探索木内に存在するか否かを判定する。指定したデータをもつノードが存在するときは TRUE を、存在しないときは FALSE を返す。
insert	指定したデータをもつノードを2分探索木に挿入する。挿入されたノードは葉となる。指定したデータをもつノードが既に2分探索木内に存在する場合は何もしない。

2分探索木を扱うために、それぞれのノードの情報を保持するデータ構造 node を定義する。データ構造 node へは、node の場所を指す変数であるポインタによってアクセスする。node がもつ変数を表2に示す。ここで、nil は、どの node も指していないことを示すポインタである。

表 2 node がもつ変数

変数名	内容
value	ノードがもつデータ
left	自分の左側の子のノードを表す node へのポインタ 左側の子がない場合は nil
right	自分の右側の子のノードを表す node へのポインタ 右側の子がない場合は nil

node と node へのポインタの関係を図2に示す。図2では、変数 p は nodeo への

ポインタであり、また、`node0` の変数 `left`, `right` は、それぞれ `node1`, `node2` へのポインタである。

このとき、`node0` の三つの変数はそれぞれ `p->value`, `p->left`, `p->right` と表す。また、図 2 に示す状況で、`p->left` の値を `p` へ代入すると、変数 `p` は `node1`

5 へのポインタとなる。

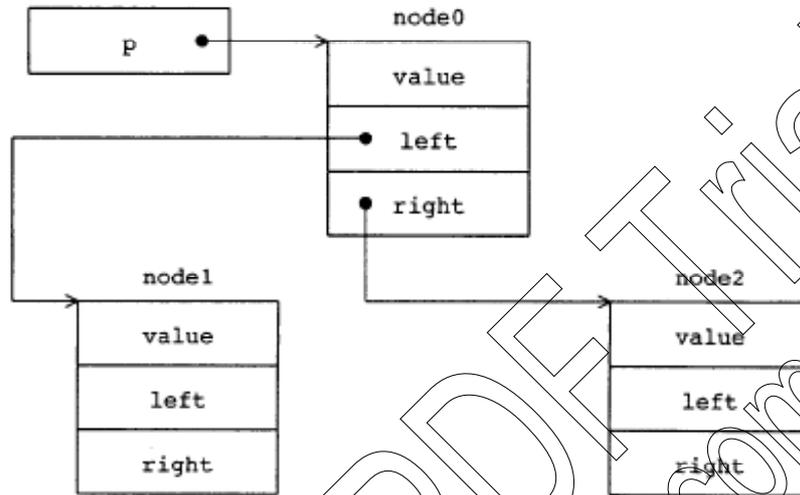


図 2 node と node へのポインタの関係

操作 `lookup` では、2 分探索木を根から葉の方向へ順次たどりながら探索を行う。まず、探索するデータと木の根がもつデータとを比べて、等しければ探索は終了する。探索するデータの方が小さければ左の子のノードに、大きければ右の子のノードに移動する。ここで、移動しようとした先に子ノードが存在しない場合、探索するデータが 2 分探索木内に見つからなかったことになり、探索終了となる。移動した先のノードでも同様にデータを比較し、探索するデータが見つかるか、又は見つからないことが分かるまで繰り返す。

10 ループを用いてポインタを順にたどる方法で操作 `lookup` を実現するプログラムを  
15 図 3 に示す。

```

function lookup( x, t )
/* x:探索するデータ, t:探索を始めるノードへのポインタ */
while ( ア )
  if ( x が t->value に等しい )
    return TRUE      /* 探索するデータが見つかり, 探索終了 */
  else
    if ( x が t->value よりも小さい )
      イ ← ウ
    else
      エ ← オ
    endif
  endif
endif
return FALSE      /* 探索するデータが見つからず, 探索終了 */
endfunction

```

図 3 ループを用いてポインタを順にたどっていく操作 lookup のプログラム

- 図 3 では、操作 lookup を、ループを用いてポインタを順にたどる方法で実現しているが、再帰呼出しを使って操作 lookup を実現する方法も考えられる。その実現方法を図 4 に示す。

```

function lookup( x, t )
/* x:探索するデータ, t:探索を始めるノードへのポインタ */
if ( t が nil である )
  return FALSE      /* 探索するデータが見つからず, 探索終了 */
endif
if ( カ )
  return TRUE      /* 探索するデータが見つかり, 探索終了 */
else
  if ( x が t->value よりも小さい )
    return lookup( x, キ )
  else
    return lookup( x, ク )
  endif
endif
endfunction

```

図 4 再帰呼出しを使った操作 lookup のプログラム

- 操作 insert のプログラムを図 5 に示す。insert は lookup と同様に、木を根から葉の方向へ進みながら、指定されたデータをもつノードを挿入する適切な位置を探す。ノードを挿入する適切な位置とは、あるノード（これを“A”とする）から移動すべき先のノードへのポインタが nil になり、どのノードも指さなくなる場所である。この時点で、指定されたデータをもつノードを新たに作成し、ノード A がもつ移動すべき先へのポインタが、新たに作成したノードを指すようにする。新たに作成されたノードは、この時点では子のノードをもたない。図 5 のプログラムでは、戻り値

として、新たにノードが挿入された場合は挿入されたノードへのポインタを返し、ノードが挿入されなかった場合には引数で渡されたポインタをそのまま返す。

```
function insert( x, t )
/* x:挿入するデータ, t:挿入を始めるノードへのポインタ */
if ( tがnilである )
 新たに node を作成し, node へのポインタを t に代入
  t->value ← ケ
  t->left  ← コ
  t->right ← コ
else
  /* tがnilでない */
  if ( xがt->valueよりも小さい )
    サ ← insert ( x, t->left )
  else
    if ( xがt->valueよりも大きい )
      シ ← insert ( x, t->right )
    endif
  endif
endif
return t
endfunction
```

図5 操作insertのプログラム

- 5 設問1 図1の2分探索木に、操作insertを使って40, 34, 38の順にデータを挿入したときの2分探索木を、解答欄に示せ。
- 設問2 図3, 図4の操作lookupのプログラム中の  ~  に入れる適切な字句を答えよ。
- 設問3 図5の操作insertのプログラム中の  ~  に入れる適切な字句を答えよ。
- 10 設問4 次の記述中の  ,  , に入れる適切な字句を答えよ。

2分探索木では、データの探索時にノードの配置によってデータを比較する対象となるノードの個数が変わる。 $n$  ( $n = 2^d - 1$ ,  $d$ は0以上の整数) 個のノードがある場合、データを比較する対象となるノードの個数は、最も多い場合  個となる。しかし、根からどの葉に向かって木をたどっても、途中で通るノードの数が同じである完全2分木の場合、データを比較する対象となるノードの個数は最も多くても  個となる。

問 6 受注管理システムの構築に関する次の記述を読んで、設問 1～4 に答えよ。

情報システムを構築する場合、これまでは端末となる各 PC に専用アプリケーションソフトをインストールし、データベース (DB) サーバなどと接続する“クライアントサーバ方式”を利用していた。これは PC の性能が向上したために、ユーザインタフェースや一部のデータ処理をクライアント側で担当し、サーバ側の処理負荷を軽減させるために有効な方式であった。しかし、運用上不都合な点が多いので、最近では PC 側のアプリケーションソフトとして汎用の Web ブラウザを用いる“Web-DB 連携システム方式”でシステム構築する例が増えている。

Web-DB 連携システムは、Web サーバ、アプリケーションサーバ (AP サーバ)、DB サーバの 3 層構造で作られることが多い。通常、Web サーバではユーザインタフェース、AP サーバではビジネスロジック (アプリケーションで実現すべき機能)、DB サーバではデータの管理を担当する。

A 社ではこれまで、専用線を用いたクライアントサーバ方式の受注管理システムを使用して、全国 100 か所の販売店から商品を受注していた。今回、受注管理システムのリプレースに際し、IP-VPN を用いた Web-DB 連携システム方式で再構築することにした。A 社の新しい受注管理システムの構成は図 1 のとおりである。また、DB 上のテーブルとテーブル間の関連は、図 2 に示すとおりである。

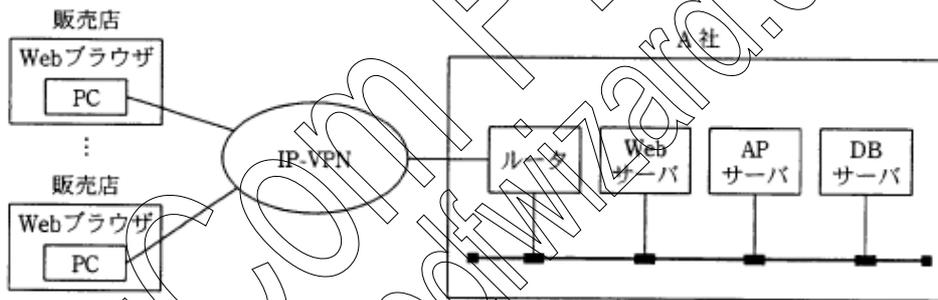


図 1 A 社の新しい受注管理システムの構成

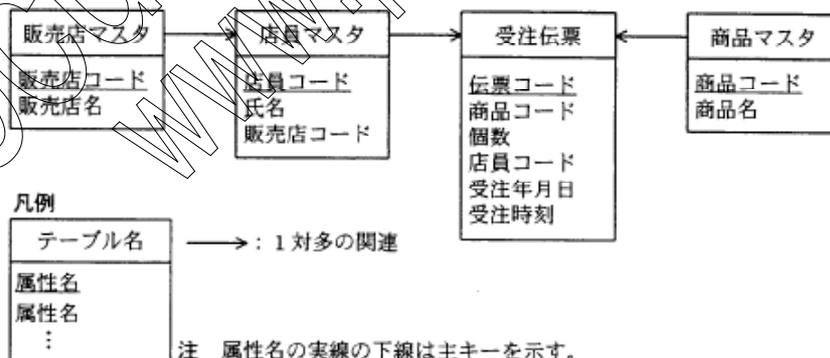


図 2 A 社の新しい受注管理システムのテーブル構造

20

設問 1 クライアントサーバ方式と Web-DB 連携システム方式を比較した場合、本文中の下線部のクライアントサーバ方式の“運用上不都合な点”に該当するものを解答群の中から二つ選び、記号で答えよ。

解答群

- ア DB の内容に不整合が起こりやすい。
- イ OS やミドルウェアなどの PC 環境を統一しないと、専用アプリケーションソフト
- 5 トが動かない場合がある。
- ウ 業務機能追加の際に、専用アプリケーションソフトを PC にインストールし直す必要がある。
- エ 業務に特化した機能を実現しにくい。
- オ サーバに負荷がかかりやすいので、クライアント数が増加したときはサーバの
- 10 能力がすぐに足りなくなってしまう。
- 力 頻繁に PC をリプレースしなければならない。

設問 2 A社では、最近の受注伝票を簡易に閲覧するために、指定した販売店で3日前

15 以降に発行された受注伝票の商品コード、商品名、個数の3項目を、受注が最も

新しいものから順に一覧表示する画面を作成した。その画面に表示する情報を取り出すために作成した、次の SQL 文中の  ~  に入れる適切な字句を答えよ。ただし、指定した販売店コードはホスト変数“:指定販売店

コード”, 今日の日付はホスト変数“:今日”, 3 日前の日付はホスト変数“:表示

20 開始日”にあらかじめ格納されているものとする。

```

SELECT 商品マスタ.商品コード, 商品マスタ.商品名, 受注伝票.個数
FROM 受注伝票, 商品マスタ, 店員マスタ, 販売店マスタ
WHERE 受注伝票.商品コード=商品マスタ.商品コード
AND 受注伝票.店員コード=店員マスタ.店員コード
25 AND 店員マスタ.販売店コード=販売店マスタ.販売店コード
AND 
AND 
AND 受注伝票.受注年月日<=:今日


```

30 設問 3 A社では、前日の受注伝票から毎日の夜間バッチ処理で、“販売店コード”, “商品コード”, “販売店名”, “商品名”, “受注年月日”, “個数”の6個のフィールドをもつ“受注集計”テーブルを作成し、受注日報作成の高速化を図った。

“受注集計”テーブルに必要なレコードを追加するために作成した、次の SQL

35 文中の  ~  に入れる適切な字句を答えよ。ただし、作成する前日の年月日はホスト変数“:前日年月日”に格納されているものとする。

INSERT INTO

```

受注集計 (販売店コード, 商品コード, 販売店名, 商品名, 受注年月日,
         個数)
40 SELECT 販売店マスタ.販売店コード, 商品マスタ.商品コード,

```

販売店マスタ.販売店名, 商品マスタ.商品名, : 前日年月日,

FROM 受注伝票, 商品マスタ, 店員マスタ, 販売店マスタ

WHERE 受注伝票.商品コード=商品マスタ.商品コード

5 AND 受注伝票.店員コード=店員マスタ.店員コード

AND 店員マスタ.販売店コード=販売店マスタ.販売店コード

AND

GROUP BY

- 10 設問4 受注管理システムの運用を開始してからしばらくして、販売店間で店員の人事異動があった。その結果、受注管理システムで、人事異動前に発行された受注伝票が人事異動後の販売店名で表示されてしまうという不具合が発生した。A社ではこの問題を解決するために、テーブルに属性を追加して対処することにした。属性を追加するテーブル名と追加すべき属性名を、それぞれ答えよ。

15