大阪大学大学院情報科学研究科 2013 年度 過去問解答

渡邉　航平

平成 27 年 6 月 15 日

1　【必須問題】アルゴリズムとプログラミング

1.1

(ア)　$i--> 0$, (イ) $data[-(i-MAX+1)]$ [1)]

1.2

1.2.1

$(1,b),(2,d),(3,c),(4,h),(5,a),(6,e),(7,g),(8,h)$

1.2.2

(a) 6 回 , (b) 13 回 , (c) 7 回 , (d) 10 回

1.2.3

(エ) $a_i \geq a_j$, (オ) k

1.3

不安定である．１５行目の条件式より,「左側の $key \geq$ 右側の key」とあるので，キーの値が同値であっても，入替が発生してしまい，本来の順番が守られない．

[1)]「先頭から順に」に注意

2　【必須問題】計算機システムとシステムプログラム

2.1

2.1.1

アドレス (8 ビット) : 郡内ブロック番号 + 群番号 + ブロック内アドレス
群番号 : セット数 $2 = 2^1$ より，1 ビット
ブロック内アドレス : ブロックサイズ $4 = 2^2$ より，2 ビット
郡内ブロック番号 : アドレスが 8 ビットなので，$8 - 1 - 2 = 5$ より，5 ビット

アドレス	郡内ブロック番号	群番号	ヒット	セット 0	セット 0	セット 1	セット 1
01011 0 11	11	0		**11**			
01001 0 10	9	0		**9**	11		
01001 0 01	9	0	hit	**9**	11		
00010 1 01	2	1		9	11	**2**	
01101 0 01	13	0		**13**	9	2	
00010 1 00	2	1	hit	13	9	**2**	
01101 0 01	13	0	hit	**13**	9	2	
01001 0 10	9	0	hit	**9**	13	2	
01011 0 10	11	0		**11**	9	2	
00010 1 01	2	1	hit	11	9	**2**	

表 1: 実行表

(1) 4 回 [2] , (2) 2 回 [3] , (3) 50 %

2.1.2

時間的局所性 : 最近参照されたワードが再度アクセスされる確率が高い.
空間的局所性 : 最近アクセスされたワードの近辺がアクセスされる確率が高い.

2.1.3

ブロックサイズが小さい場合，セット数ブロック内に参照したいワードがない確率が高く，ブロック置き換えが頻発するので，平均メモリアクセス時間が大きくなる.
ブロックサイズが大きい場合，自由度が高くなりブロック内での検索が困難になり，平均メモリアクセス時間が大きくなる.

2.2

2.2.1

(a) サ , (b) ク , (c) オ , (d) キ , (e) コ , (f) イ , (g) ス

[2] 1, 2, 4, 5 番目
[3] 5, 9 番目

2.2.2

(1)
(到着順) : 20, 52, 52, 76
(処理時間順) : 20, 92, 12, 36

(2)

(3)

タイムスライスが小さいと，プロセスの切り替えが頻繁に発生し，そのための処理時間が増大するため，平均ターンアラウンド時間も増大する．
また，タイムスライスが大きいと，処理時間が短いプロセスの待ち時間が長くなるため，平均ターンアラウンド時間が増大する．

3 【選択問題】離散構造

3.1

(a) 恒真

(b) 充足可能,
真：$p(a) = tt, p(b) = tt$, 偽:$p(a) = tt, p(b) = ff$

(c) 充足不能

(d) 充足可能,
真:$p(a,b) = tt, p(b,a) = tt, p(a,a) = tt, p(b,b) = tt$,
偽:$p(a,b) = tt, p(b,a) = ff, p(a,a) = ff, p(b,b) = tt$

3.2

3.2.1

$$\begin{aligned}\neg E &= A \wedge B \wedge C \wedge \neg D \\ &= [p(f(g(f(g(g(a))))))] \wedge \forall x[\neg p(f(g(x))) \vee p(x)] \wedge \forall x[\neg p(g(f(x))) \vee p(x)] \wedge [\neg p(g(x))] \\ &= \forall x\{[p(f(g(f(g(g(a))))))] \wedge [\neg p(f(g(x))) \vee p(x)] \wedge [\neg p(g(f(x))) \vee p(x)] \wedge [\neg p(g(x))\}\end{aligned}$$

これは存在限量子を含まないため，E' となる.

3.2.2

前問の [ ] を左から，(1),(2),(3),(4) とする.

(1), (2) より, $x \leftarrow f(g(g(a)))$ として, $p(f(g(g(a))))$ ... (5)
(2), (5) より, $x \leftarrow g(a)$ として, $p(g(a))$ ... (6)
(4), (6) より, $x \leftarrow a$ として，充足不能

したがって，E' は充足不能

3.3

3.3.1 反射性

$\forall a$ に対し，$(a,a) \in R$ より，$\forall a(aRa)$ なので，反射性は成立.

3.3.2 反対称性

(イ) $\forall a \in X, \forall b \in Y(a,b) \in R$ だが, $(w,z) \notin (R)$

(ロ) $\forall a \in X(a,a) \in R, a = a$ かつ $\forall b \in Y(b,b) \in R, b = b$

(イ)，(ロ) より，反対称性は成立.

3.3.3 推移性

(ハ) $\forall a = b \in X, \forall c \in Y[(a,b) \in R, (b,c) \in R]$ より, $(a,c) \in R$

(ニ) $\forall a \in X, \forall b = c \in Y[(a,b) \in R, (b,c) \in R]$ より, $(a,c) \in R$

(ホ) $\forall a \in X \wedge \neg Y, \forall c \in Y \wedge \neg X, \forall b \in X \wedge Y[(a,b) \in R, (b,c) \in R]$ より, $(a,c) \in R$

(ハ),(ニ),(ホ) より，推移性は成立.
これらの議論より，反射性，反対称性，推移性が成立するため，R は順序関係.

3.4

3.4.1

$|L_n(a)| = |L_{n-1}|, \quad |L_n(b)| = |L_{n-2}|$

3.4.2

$$|L_n| = \begin{cases} |L_{n-1}| + |L_{n-2}| & (n > 2) \\ 3 & (n = 2) \\ 2 & (n = 1) \end{cases}$$

4　【選択問題】ネットワーク

4.1

4.1.1

$C_0(010) = \phi, \quad C_1(101) = 111$

4.2

復号失敗とする

4.3

シンドローム

4.4

符号語 v と，$\Delta(v,u) \leq t$ となる語 u，$\Delta(v,u) = d$ となる符号語 w を考える．
$\Delta(w,v) \leq t$ とすると，三角不等式より，

$$\begin{aligned} d &< t + t \\ d &< 2t \end{aligned}$$

となり，定義と矛盾．
したがって，$C_t(v)$ の要素数は高々 1 つ．

4.5

生成行列 $G = (1 \quad 1 \quad 1)$ とすると，$G = [I_1 \quad A^T]$ より，$I_1 = (1), \quad A^T = (1 \quad 1)$

したがって，$H = (A \quad I_2) = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$

4.6

4.6.1

- 応答確認などのヘッダ情報がないため，伝送効率がよい．
- データ伝送前にハンドシェイクを行わないため，伝送時間短縮

4.6.2

宛先 IP アドレス，宛先ポート番号，送信元 IP アドレス，送信元ポート番号

4.6.3

- 通信がタイムアウトしたことで，コネクションを再構築する際に，同じシーケンス番号でコネクション要求を行うと，以前のコネクションに対するパケットと新しいコネクションに対するパケットが区別できなくなり，TCP プロトコルスタックが不正動作する恐れがある．シーケンス番号をランダムにすることで，コネクションの区別ができるようになる．
- 同じエンドの別プロセスから同時にコネクション要求が届いた場合，同じシーケンス番号だとそれらの区別ができない．したがって，シーケンス番号を異なるものとし，コネクションの区別を行う．

4.6.4

セグメント 1 個では，$1000-20=980bytes$ のデータ送信が可能．したがって，100100bytes のデータを運ぶためには，103 個のセグメントに分割する必要がある [4]．また，セグメント送信以前に 3-handshake でプロセス A は 2 回，プロセス B は 1 回セグメントを送信しているので，プロセス A は 104 回，プロセス B は 103 回のセグメント送信が必要．

[4] $100100 \div 980 = 102$ 余り 140