1 (1)

(1-1)

CPUを除く装置の故障率は、

$$\frac{1}{1,000,000}+\frac{2}{500,000}+\frac{1}{200,000}+\frac{1}{1,000,000}=\frac{11}{1,000,000}$$

よって、単一プロセッサの各計算機のMTTFは、

$$\frac{11+1}{1,000,000}=\frac{12}{1,000,000}$$ より、 $$\frac{1,000,000}{12}$$ [時間]。

また、8ウェイSMPの各計算機のMTTFは、

$$\frac{11+8}{1,000,000}=\frac{19}{1,000,000}$$ より、 $$\frac{1,000,000}{19}$$ [時間]。

(1-2)

$$\frac{6}{500,000}+\frac{1}{200,000}+\frac{1}{1,000,000}=\frac{18}{1,000,000}$$ より、

$$\frac{1,000,000}{18}$$ [時間]

(1-3)

図より、単一プロセッサクラスタ、8ウェイSMPクラスタの構成はそれぞれ、

単一プロセッサクラスタ		8ウェイSMPクラスタ	
計算機	32	計算機	4
Ethernetスイッチ	2	ディスクサブシステム	4
Ethernetケーブル	36	Ethernetスイッチ	1
		Ethernetケーブル	5
		SCSIケーブル	4

(1-1), (1-2)で求めた故障率を用いると、

単一プロセッサクラスタのMTTFは

$$\frac{12}{1,000,000}\times 32+\frac{2}{500,000}+\frac{36}{1,000,000}=\frac{424}{1,000,000}$$ より、 $$\frac{1,000,000}{424}$$ [時間]

NO.

DATE

8ウェイSMPクラスタのMTTFは、

$$\frac{19}{1,000,000}\times4+\frac{18}{1,000,000}\times4+\frac{1}{500,000}+\frac{5}{1,000,000}+\frac{4}{1,000,000}=\frac{159}{1,000,000}$$ より、

$$\underline{\frac{1,000,000}{159}\ [時間]}$$

(1-4)

単一プロセッサクラスタ：N1, N2

8ウェイSMPクラスタ：M2, M4, M5, D2, D3, N1, N2, N4, N5

(2)

512 ÷ 32 = 16 より、キャッシュメモリ内に作られるブロックの数は16コ

32 ÷ 4 = 8 より、1ブロックに含まれるint型要素の数は 8コ

(2-1)

直接マップ方式では、主記憶上の要素は順に16の群に分けられることになる。
8×16=128より、a[0]～a[7]を群1のブロックと考えれば、a[128]～a[135]は同じく群1のブロックであり、a[136]～a[143]は次の群2のブロックである。
K=135のとき、a[0]とa[135]に交互にアクセスすることになる。
この2つは同じ群の違うブロックにあたるため、毎回キャッシュミスが生じ、
キャッシュミス率は 100%、となる。
K=136のとき、a[0]とa[136]に交互にアクセスすることになるが、この2つは別の群なため、初回アクセス時(i=0の時)しかキャッシュミスは生じない。
よって、キャッシュミス率は 1%、となる。

(2-2)

2ウェイ群連想方式では、ブロック枠を2行×8列に構成するため、主記憶上の要素は順に8の群に分けられることになる。
8×8=64より、同様に、a[0]～a[7]を群1のブロックと考えれば、a[128]～a[135]は同じく群1のブロックであり、a[136]～a[143]は次の群2のブロックである。
K=135のとき、a[0]とa[135]に交互にアクセスすることになる。この2つは同じ群の違うブロックにあたるが、同じ群内のデータは2ブロックまで同時にキャッシュに置いておけるため、初回(i=0のとき)しかキャッシュミスは生じず、キャッシュミス率は 1%、となる。
K=136のときは、(2-1)と同様に、1% となる。

NO.

DATE

$\boxed{3}$

(1) A[1].key : 10　A[2].key : 18　A[3].key : 20　A[4].key : 35

A[5].key : 75　A[6].key : 44　A[7].key : 4　A[8].key : 12.

(2) 最大値：11〜19行目のループ内で、最大 i 回、16行目の比較を行う。

よって、$\sum_{i=2}^{N} i = \sum_{i=1}^{N} i - 1 = \frac{N(N+1)}{2} - 1 = \underline{\frac{1}{2}N^2 + \frac{1}{2}N - 1 \text{ (回)}}$

最小値：11〜19行目のループ内で16行目の比較は最小の場合、1度行うだけですむ。

よって、$\underline{N-1}$ (回)

(3) この手続きのうち、要素入れ替えの部分は16,17行目にあたる。16行目の比較において、w.key = A[j].key の場合は条件にあたらないため、入れ替えは行われない。よって、手続き sort は安定である。

(4) j を1ずつ減らしていき、j=0となった時点でループを抜けるようにしなくてはならない。

よって、<u>(j > 0) and (w.key < A[j].key)</u>

$\boxed{4-A}$

(A1) $I_1 : p(a,b) \Rightarrow a \leqq b$

・すべての非負整数 x に対して、$x \leqq x$

・すべての非負整数 x, y, z に対して　$x \leqq y$ かつ $y \leqq z$ ならば $x \leqq z$

が言え、このとき、すべての非負整数 y に対して $0 \leqq y$ が言えるため、与えられた論理式は真となる。

$I_2 : p(a,b) \Rightarrow a = b$

・すべての非負整数 x に対して、$x = x$

・すべての非負整数 x, y, z に対して　$x = y$ かつ $y = z$ ならば $x = z$

が言え、このとき、すべての非負整数 y に対して $x = y$ となるような x は存在しないため、論理式は負となる。

NO.

DATE

(A2) 与えられた論理式をHとすると、

$\neg H = \forall x \forall y (\neg p(x,y) \vee p(y,x)) \wedge \forall x \forall y \forall z (\neg p(x,y) \vee \neg p(y,z) \vee p(x,z))$

$\wedge \forall y \exists x\, p(x,y) \wedge \exists x (\neg p(x,x))$

$= \exists u \forall y \exists v \forall x \forall z ((\neg p(x,y) \vee p(y,x)) \wedge (\neg p(x,y) \vee \neg p(y,z) \vee p(x,z))$

$\wedge p(v,y) \wedge \neg p(u,u))$

$u=a$, $v=f(y)$としてスコーレム関数を導入し、導出節を求めていく。

$\neg p(x,y) \vee p(y,x)$ (1)

$\neg p(x,y) \vee \neg p(y,z) \vee p(x,z)$ (2)

$p(f(y),y)$ (3)

$\neg p(a,a)$ (4)

(2)で$x=a$, $y=f(a)$, $z=a$とおくと、 $\neg p(a,f(a)) \vee \neg p(f(a),a) \vee p(a,a)$ (5)

(4)(5)のリゾルベント $\neg p(a,f(a)) \vee \neg p(f(a),a)$ (6)

(3)で$y=a$とおくと、 $p(f(a),a)$ (7)

(6)(7)のリゾルベント $\neg p(a,f(a))$ (8)

(1)で$x=f(a)$, $y=a$とおくと、 $\neg p(f(a),a) \vee p(a,f(a))$ (9)

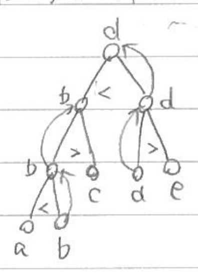
(8)(9)のリゾルベント $\neg p(f(a),a)$ (10)

(7)(10)のリゾルベントより 空節が導き出せるため、$\neg H$は充足不能であり、Hが恒真であることが言える。

(A3) [a] $\forall y (p(x,y) \rightarrow q(y))$ [b] $\exists x (q(x) \wedge p(x,y))$

4-B

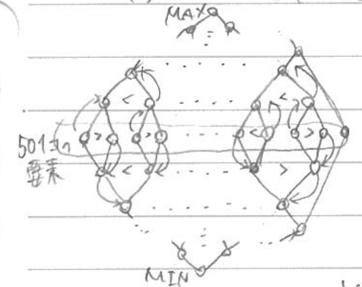
(B1) 上位インデックスの1つを見つけるためには、その数より小さい数が少なくともn個あるということを言う必要がある。これは適当に選んだn+1個以上の数の最大値を見つける問題と言うことができる。



最大値を見つけるためには、ある集団の中で最大の数と、別の集団の中で最大の数を比較し、より大きいものを取り出す、といった作業を繰り返すことになる。比較は2つの数に対して行われるので、この作業は左のような2分木に書くことができる。

このとき、比較の回数は、葉でない節点の数で表せる。
また、n+1個以上の数を比較する必要があるため、葉の数はn+1個以上である。よって、葉でない節点の数はn以上となり、「Sの上位インデックスの1つをn−1回以下の比較により確実に見つける方法は存在しない」と言える。

(B2) こちらも同じく、n+1=501個の要素の最大値と最小値を求める問題を考えることができる。



左図のように、今度は大きいもの同士、小さいもの同士で比べ合うことを考える。
具体的には、まず任意のただしすべて異なる要素を用いた250対を比較する。次に、より大きかったものから125対、より小さかったものから125対を選び比較する。
その次にまた、62+1対（+1は、125のうち対にしそこねたものと最初に対にしなかった、501のうちの1つを対にしたもの）ずつを比較する…ということを最大値、最小値が出るまで繰り返す。これは、左のように上下にのびる木として表せる。
この時、片方の木の葉でない節点数は500。最初の250対の比較時のみ、最大値導出、最小値導出に共通で行われるため、

$$500 \times 2 - 250 = 750$$

回の比較で上位インデックスのうちの1つ、下位インデックスのうちの1つを求められる。

II

$\boxed{1}$ (1)

(1-1) ア　wait → signal の順に用いる。

(1-2) (E) P(S)　(F) V(M)　(G) P(M)　(H) V(S)

(1-3) (1-2)の問題で、N=1のときを考える。
プロデューサのP命令の途中でコンシューマのV命令が割り込んだ場合、以下のようになることが考えられる。
このとき、S側の待ち行列をQ、M側の待ち行列をRとする。

プロデューサ	コンシューマ	s	m
P(S)	P(M)	1	0
→S←S−1して次へ	→プロセスを待ち状態に	0	0
V(M) →	→待ち状態解除		
P(S)	V(S)		
if(S>0):S=0よりfalse	if(Qが空ではない) 空なのでfalse		
Qにプロセスを登録、待ち状態に	割り込み S←S+1	1	0
	P(M)		
	→m=0なので プロセスを待ち状態に		

結果として、両方のプロセスが待ち状態となり、デッドロックの状態になってしまう。上のV(S)命令のif文直後にP(S)命令が割り込んだ場合でも同じことが起きる。
以上より、P命令、V命令は実行途中に他のプロセスによって割り込まれることがあってはならない。

(2)

(2-1) 2KB：128÷2=64より、 10×64+128+1.28=769.28≒769 msec.
16KB：128÷16=8より、 10×8+128+1.28=209.28≒209 msec.

(2-2) 2KB： 2+2+2+4+10+12+130+250=412 KB
16KB： 16×6+144+256=496 KB.

(2-3) ブロックサイズを大きくすると、シーク時間と回転待ち時間を減らすことができ、読み出し時間が短くなる。これが利点である。
欠点は、ブロック単位でしかデータを扱えないため、余分な領域が生まれ、ディスク利用率が上がってしまうことである。

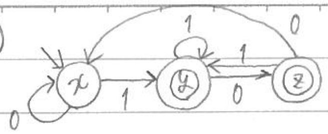
$\boxed{2}$ (1)

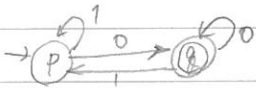
(1-1) $1^*0(1^*0)^*$ ：01が途中何度入るうが、最後は0で終わる。

NO.

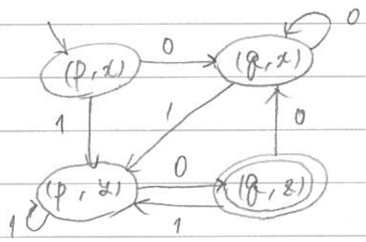
DATE

(1-2)



(1-3) (1-1)の状態名を  とすると、オートマトンM_3の状態遷移は.

	0	1	受理
(p,x)	(q,x)	(p,y)	
(q,x)	(q,x)	(p,y)	
(p,y)	(q,z)	(p,y)	
(q,z)	(q,x)	(p,y)	○



(1-3-1)

10, 010, 110

(1-3-2)

4で割り切れない偶数

(2)

(2-1) a, a*a, a+a, (a), (a*a), (a+a), a+a*a, a*a+a, ((a)),
a*a*a, a+a+a, a+(a), (a)+a, a*(a), (a)*a

(2-2)

(S, S+S, A+S, a+S, a+S+S, a+A+S, a+A*A+S, a+a*A+S, a+a*a+S,
a+a*a+A, a+a*a+a)

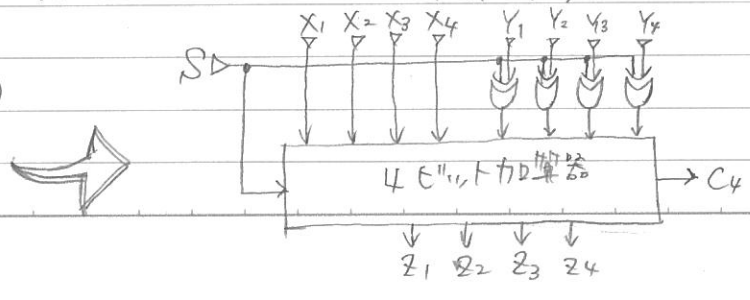
(2-3)

(S, S+S, S+A, S+a, S+S+a, S+A+a, S+A*A+a, S+A*a+a, S+a*a+a,
A+a*a+a, a+a*a+a)

(2-4) 1. a 2. A*A 3. (S+S)

$\boxed{3}$ (1)

(1-1)



NO.

DATE

(1-2)

(1-2-1) $C_i = G_i + P_i C_{i-1}$

(1-2-2)

(ア) $X_1 \cdot Y_1 + (X_1 + Y_1) \cdot C_0$

(イ) $X_2 \cdot Y_2$ (ウ) $(X_2 + Y_2) \cdot (X_1 + Y_1)$

(エ) $X_3 \cdot Y_3$ (オ) $(X_3 + Y_3) \cdot (X_2 + Y_2) \cdot (X_1 + Y_1)$

(カ) $X_4 \cdot Y_4$ (キ) $(X_4 + Y_4) \cdot (X_3 + Y_3) \cdot (X_2 + Y_2) \cdot X_1 \cdot Y_1$

(ク) $(X_4 + Y_4) \cdot (X_3 + Y_3) \cdot (X_2 + Y_2) \cdot (X_1 + Y_1)$

(2)

(2-1) (ア) S_0 (イ) S_1 (ウ) S_0 (エ) S_1 (オ) S_3 (カ) S_0

(キ) 0 (ク) 0 (ケ) 0 (コ) 1.

(2-2) $Y = Q_3$.

(2-3)

D_1

Q_1Q_2 \ Q_3X	00	01	11	10
00	1			1
01	1			1
11	1	d		1
10	1			1

D_2

Q_1Q_2 \ Q_3X	00	01	11	10
00				1
01	1			1
11		d		1
10		1		1

D_3

S4になるときのみ1,
他の場合0なので,
S_3かつXつまり
$D_3 = Q_1 \cdot Q_2 \cdot \overline{Q_3} \cdot X$.

$D_1 = \overline{X}$

$D_2 = Q_3 \cdot \overline{X} \vee Q_1 \cdot \overline{Q_3} \cdot X \vee \overline{Q_1} \cdot Q_2 \cdot \overline{X}$