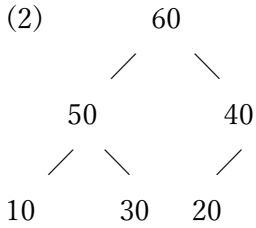
大問 1 アルゴリズムとプログラミング(解答作成者：Hen)

(1) ヒープソート

(2)



(3) 節点番号が current のデータはその子のデータより大きい

(4) 最悪時間量:　O(nlogn)

(5)

(5-1)

あ：n/2　　い：d　　う：n　　え：i

(5-2)

(6)

あ：n　　い：d　　う：i　　え：i/2

大問２：計算機システムとシステムプログラム(解答作成者：楊)

(1)

(1-1)(a)エ　(b)イ　(c)ウ　(d)ア　(e)ク

(1-2)

(1-2-1)

S=0，e=14，

m:111111111，1+m=1.111111111

$A=1.111111111*2^{7}=(11111111.11)_2$

十進制：255.75

(1-2-2)

S=0，e=1，

m:000000000，1+m=1.000000000

$A=1.000000000*2^{-6}=(0.000001)_2$

十進制：0.015625

(1-2-3)

$(36.66)_{10}=(100100.101010001111010111000010100011110101110000101)_2$

1+m=1.001001010

$e=5+7=(12)_{10}=(1100)_2$

s=1

ビット列：11100001001010

(1-2-4)丸め処理として切り捨てを用いると、ほかの丸め処理方式より、誤差が一番大きいである.

(2)

(2-1)(a)キ　(b)カ　(c)オ　(d)ア　(e)コ　(f)ソ　(g)ス　(h)エ　(i)ア

(2-2)

	0	1	2	0	3	1	4	3	2	3	1	2	4
Page frame1	0	1	2	2	3	3	4	4	4	4	1	2	2
Page frame2		0	1	1	2	2	3	3	3	3	4	1	1
Page frame3			0	0	1	1	2	2	2	2	3	4	4
Page fault	○	○	○		○		○				○	○	

(2－3)

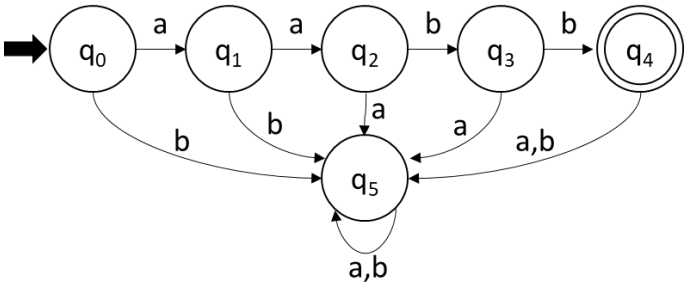
メモリアクセス総回数：A

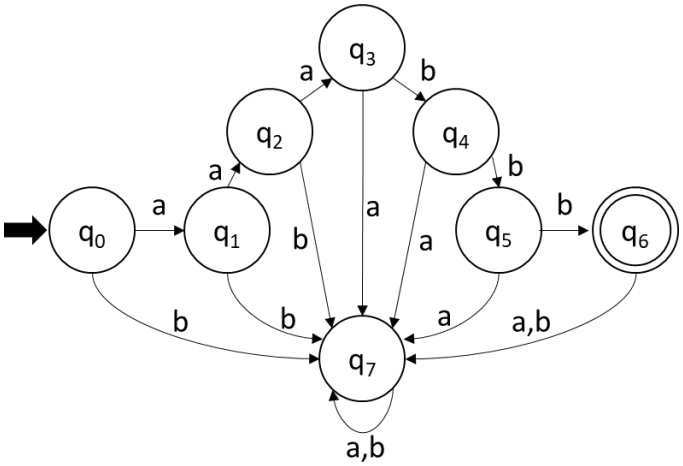
$AP(8ms+4\mu s)/4A\mu s \leq 0.1$

大問 4 計算理論(解答作成者：松井)

(1-1) (A) (a,0)/00　(B) (b,0)/ ε　(C) (b,0)/ ε

(1-2-1)



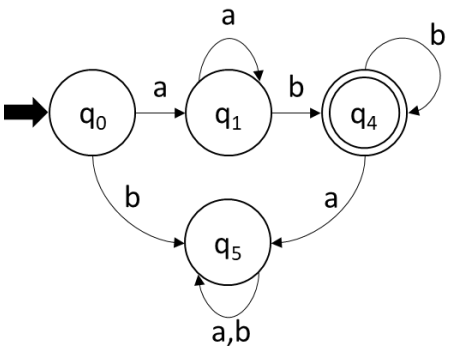


(1-2-2)

k+1 個の $p_i(i=0,1,...,k)$がすべて相異なるということはありえない．したがって二つの異なる整数 f と g(0≦f<g≦k)で $p_f=p_g$ を満たすものが存在する．そこで文字列を $xyzb^k(x=a^f, y=a^{g-f}, z=a^{k-g})$と分解すると，$p_f=p_g$ から，xy^2zb^k もオートマトンは受理する．しかし，xy^2zb^k は $a^{k+g-f}b^k$ であり，言語 L_{ab} の要素ではない．

↑正直なところ，もっとスマートな解き方がありそう

(1-3)



(2-1)

M[i,j]	j=1	j=2	j=3	j=4
i=1	{A,C}	{A}	{A}	{S}
i=2	—	{A,C}	{A}	{S}
i=3	—	—	{A,C}	{S}
i=4	—	—	—	{B,D}

(2-2)

文字列 aa について考える．S→SA→SAA→AA→aA→aa から，文字列 aa は G_2 によって生成されることがわかる．しかしこの文字列にアルゴリズムを適用すると，結果は以下の表のようになり，生成されないと判断され，正しく動作していないことがわかる．

M[i,j]	j=1	j=2
i=1	{A}	Φ
i=2	—	{A}

大問 5（解答作成者：福家）
(1)
(1-1)再送要求
(あ)send(0, data)
(い)Λ

(1-2)
(1-2-1)受信側が応答パケットを正しく送信できているか確認できない、かつ受信したデータが何番目かを把握できない場合、同一パケットを受信することによりファイル内に同じ内容が入ってしまう。

(1-2-2)これで受信側が正しく応答パケットを出せたか分かる。
(う)OnRecvNack
(え)OnRecvBitter
(お)sendData(0, data)
(か) OnRecvBitter
(き) sendData(1, data)

(1-3)最大なので応答パケットのビット誤りなし
1 パケットあたり(24/1600(s)+10(ms))+(8/1600(s)+10(ms))
よって 25 パケット

(1-4)
パケットロスが起きた場合を考えて、送信端末は受信端末から一定時間 ACK が返って来なければビット誤り応答パケット受信時と同様にパケットを再送する。なお、このパケットロスは送信時・受診時どちらに起きても問題ない。

(2)
(2-1)
(あ)パリティ
(い)2
(う)1
(え)バースト誤り
(お)r←(分かりやすい説明)

(2-2)

(2-2-1)

M(x)に x^r を掛けて　G(x)で割った剰余に、M(x)に x^r を掛けたものを足す。

(2-2-2)

G(x) = x^4 + x + 1

M(x) = x^7 + x^6 + x^4 + x + 1

F(x) = x^11 + x^10 + x^8 + x^5 + x^4 + x^3 + 1

(2-2-3)

受信多項式を生成多項式で割り、余りで誤り位置を調べる。（参考サイト）

大問 6(解答作成者：栗原)

(1)未回答

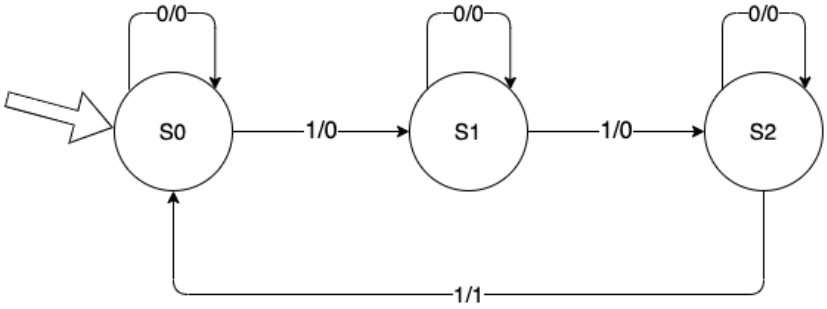
(1-1)

(1-2)

(1-3)

(2)

(2-1) 入力/出力 x_0/y_0



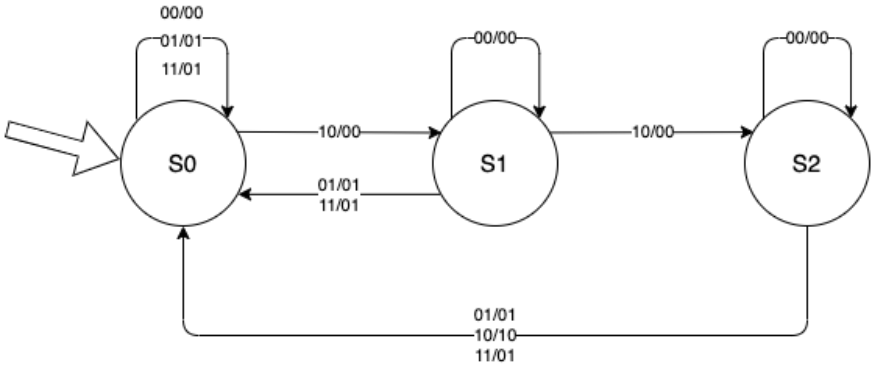
(2-2)

$$k_0^+ = \ k_0 \wedge \neg x_0 \vee \neg k_1 \wedge x_0$$
$$k_1^+ = \ k_1 \wedge \neg x_0 \vee \neg k_0 \wedge \neg k_1 \wedge x_0$$
$$y_0 = \ k_0 \wedge x_0$$

(2-3)論理式

$$k_0^+ = \ k_0 \wedge \neg x_0 \wedge \neg x_1 \vee \neg k_1 \wedge x_0 \wedge \neg x_1$$
$$k_1^+ = k_1 \wedge \neg x_0 \wedge \neg x_1 \vee \neg k_0 \wedge \neg k_1 \wedge x_0 \wedge \neg x_1$$
$$y_0 = \ k_0 \wedge x_0 \wedge \neg x_1$$
$$y_1 = \ x_1$$

状態遷移図: 入力/出力 x_0x_1/y_0y_1



(2-4)

機能:返金機能（リセット）

理由：$x_1 = 0$の時，状態は変化しない

$x_1 = 1$の時，どの状態であっても初期状態（状態 A）に遷移しているから．

つまり，押された時，入金されたものが返金されている(合計投入金額が 0 になっている)．