

**Q1**

(a)	(b)	(c)	(d)	(e)	(f)
A (96 Mb)	B(34 Mb)	C(18 Mb)	A leaves	D(48 Mb)	E(8Mb)
96   128	96   128	96   128	128	48   128	48   128
128	34   128	34   128	34   128	34   128	34   128
128	128	18   128	18   128	18   128	18   128
128	128	128	128	128	8   128
			<b>F Rate</b>	0,61	<b>F Rate</b> 0,70

(a)	(b)	(c)	(d)	(e)	(f)
A (96 Mb)	B(34 Mb)	C(18 Mb)	A leaves	D(48 Mb)	E(8Mb)
16	16	16	16	16	8   16
32	32	18   32	18   32	18   32	18   32
64	34   64	34   64	34   64	34   64	34   64
96   128	96   128	96   128	128	48   128	48   128
			<b>F Rate</b>	0,35	<b>F Rate</b> 0,39

The blocks that are not allocated are not considered in the fragmentation rate

The blocks that are not allocated are not considered in the fragmentation rate

With best fit and unequal partitionning, the FRate is best  
Due to the block size constraints, best fit allocation has no real impact here

**Q2**

first fit	t0	(a) P7(8)	(b) P2 leaves	(c) P3 leaves	(d) P8(22)
P1   32	P1   32	P1   32	P1   32	P1   32	P1   32
H   16	P7   8	P7   8	P7   8	P7   8	P7   8
P4   64	H   8	H   8	H   8	H   8	H   8
P5   8	P4   64	P4   64	P4   64	P4   64	P4   64
H   4	P5   8	P5   8	P5   8	P5   8	P5   8
P2   18	H   4	H   44	H   50	P8   22	P8   22
H   22	P2   18	P3   6	P6   30	H   28	H   28
P3   6	H   22			P6   30	P6   30
P6   30	P3   6				
	P6   30				
					<b>F Rate</b> 0,22

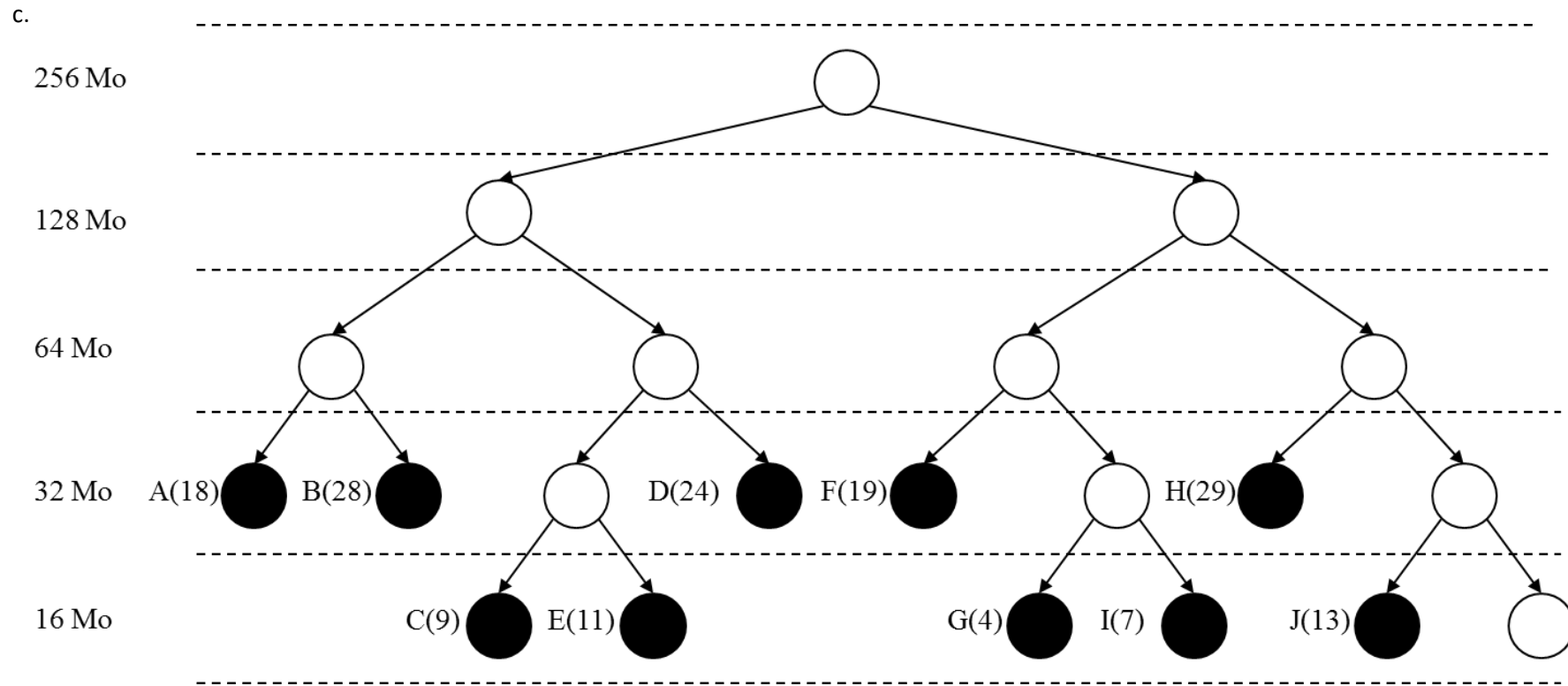
  

next fit	t0	P7(8)	P2 leaves	P3 leaves	P8(22)
P1   32	P1   32	P1   32	P1   32	P1   32	P1   32
H   16	H   16	H   16	H   16	H   16	H   16
P4   64	P4   64	P4   64	P4   64	P4   64	P4   64
P5   8	P5   8	P5   8	P5   8	P5   8	P5   8
H   4	H   4	H   4	H   22	H   22	P8   22
P2   18	P2   18	P7   8	P7   8	P7   8	P7   8
H   22	P7   8	H   14	H   20	H   20	H   20
P3   6	H   14	P3   6	P6   30	P6   30	P6   30
P6   30	P3   6	P6   30			
	P6   30				
					<b>F Rate</b> 0,44

surprisingly, first fit performs best

a double linked list with a circular chain is needed to deal with the next fit, if looping while searching

**Q3**



a. 5 levels (256, 128, 64, 32 and 16 Mb)

b. A, B, D, F and H on the fourth level (32 Mb)

C, E, G, I, J on the fifth level (16 Mb)

d, e. recursive searches are supposed to be in the [0 4] interval (tree deep)

- A 3
- B 0
- C 2
- D 0
- E 0
- F 2
- G 1
- H 1
- I 0
- J 1

	t0	A	B	C	D	E	F	G	H	I	J
<b>256</b>	1										
<b>128</b>		1	1	1	1	1					
<b>64</b>		1	1				1	1			
<b>32</b>		2	2	3	3	3	5	4	6	6	6
<b>16</b>				2	2	2	2	4	4	4	4
	256	256	256	256	256	256	256	256	256	256	256

max is 3 and obtained for A

min is 0 and obtained for B, D, E, I

f.

	Allocated	Hole	Buddy
A	18	14	32
B	28	4	32
C	9	7	16
D	24	8	32
E	11	5	16
F	19	13	32
G	4	12	16
H	29	3	32
I	7	9	16
J	13	3	16
<b>Free</b>		16	
	F Rate	0,82	

The buddies that are not allocated are not considered for the fragmentation rate i.e. only the last buddy (right) of 16 MB

**Q4**

64 Kb =  $2^{16}$  then  $d = 16$

4 Gb =  $2^{32}$  then  $m = 32$

$d = 16, m = 32$  then  $p = 16$  and we have  $2^{16} = 65536$  frames / pages

1024 entries =  $2^{10}$  pages, with a page size of  $2^{16}$  we have  $2^{10} * 2^{16} = 2^{26} = 64$  Mb

In the best case  $2^{26} / 2^{32} = 1 / 2^6 = 1/64$  of the memory can be loaded at a maximum in the TLB

- a.
- A(140) = 3 pages
  - B(96) = 2 pages
  - C(220) = 4 pages
  - D(205) = 4 pages
  - E(550) = 9 pages

b.	(init)	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
		A loaded	B loaded	C loaded	B leaves	D loaded	A leaves	C leaves	E loaded
0		A0	A0	A0	A0	A0			E0
1		A1	A1	A1	A1	A1			E1
2		A2	A2	A2	A2	A2			E2
3			B0	B0		D0	D0	D0	D0
4			B1	B1		D1	D1	D1	D1
5				C0	C0	C0	C0		E3
6				C1	C1	C1	C1		E4
7				C2	C2	C2	C2		E5
8				C3	C3	C3	C3		E6
9						D2	D2	D2	D2
10						D3	D3	D3	D3
11									E7
12									E8

c.

Page table - D

0	3
1	4
2	9
3	10

Page table - E

0	0
1	1
2	2
3	5
4	6
5	7
6	8
7	11
8	12

d.

8F0F0 corresponds to p = 8 and d = F0F0  
then it becomes CF0F0 with frame = C (i.e. 12)

e.

	Process	Pages	Holes
A	140	192	52
C	220	256	36
D	205	256	51

At F Rate =  $1 - 52 / (52 + 36 + 51) = 0,62$