Problem 1

1.

(a) FIFO

Page Access	Actions	Buffer
-	-	$\top, \top, \top$
read p2	load(p2)	$p_2, \top, \top$
	get(p2)	$p_2, \top, \top$
write p3	load(p3)	$p_2, p_3, \top$
	get(p3)	$p_2, p_3, \top$
write p2	get(p2)	$p_2, p_3, \top$
read p1	load(p1)	p_2, p_3, p_1
	get(p1)	p_2, p_3, p_1
read p0	evict()	$p_3, p_1, \top$
	load(p0)	p_3, p_1, p_0
	get(p0)	p_3, p_1, p_0
write p4	evict()	$p_1, p_0, \top$
	load(p4)	p_1, p_0, p_4
	get(p4)	p_1, p_0, p_4
write p1	get(p1)	p_1, p_0, p_4
write p3	evict()	$p_0, p_4, \top$
	load(p3)	p_0, p_4, p_3
	get(p3)	p_0, p_4, p_3
read p0	get(p0)	p_0, p_4, p_3
read p1	evict()	$p_4, p_3, \top$
	load(p1)	p_4, p_3, p_1
	get(p1)	p_4, p_3, p_1

(b) LFU

Page Access	Actions	Buffer	usage
-	-	$\top, \top, \top$	$0, 0, 0$
read p2	load(p2)	$p_2, \top, \top$	$0, 0, 0$
	get(p2)	$p_2, \top, \top$	$1, 0, 0$
write p3	load(p3)	$p_2, p_3, \top$	$1, 0, 0$
	get(p3)	$p_2, p_3, \top$	$1, 1, 0$
write p2	get(p2)	$p_2, p_3, \top$	$2, 1, 0$
read p1	load(p1)	p_2, p_3, p_1	$2, 1, 0$
	get(p1)	p_2, p_3, p_1	$2, 1, 1$
read p0	evict()	$p_2, p_1, \top$	$2, 1, 0$
	load(p0)	p_2, p_1, p_0	$2, 1, 0$
	get(p0)	p_2, p_1, p_0	$2, 1, 1$
write p4	evict()	$p_2, p_0, \top$	$2, 1, 0$
	load(p4)	p_2, p_0, p_4	$2, 1, 0$
	get(p4)	p_2, p_0, p_4	$2, 1, 1$
write p1	evict()	$p_2, p_4, \top$	$2, 1, 0$
	load(p4)	p_2, p_4, p_1	$2, 1, 1$
	get(p4)	p_2, p_4, p_1	$2, 1, 2$
write p3	evict()	$p_2, p_1, \top$	$2, 2, 0$
	load(p3)	p_2, p_1, p_3	$2, 2, 1$
	get(p3)	p_2, p_1, p_3	$2, 2, 2$
read p0	evict()	$p_1, p_3, \top$	$2, 2, 0$
	load(p0)	p_1, p_3, p_0	$2, 2, 1$
	get(p0)	p_1, p_3, p_0	$2, 2, 2$
read p1	get(p1)	p_1, p_3, p_0	$3, 2, 2$

2.

Miss ration FIFO $= \frac{7}{10} = 0.7$.

Miss ration LFU $= \frac{8}{10} = 0.8$

Problem 2

1.

Compression only make sense if $T_{trc}(n) < T_{truc}(n)$.

In order to determine the minimal compresion speed, we need to first compute $T_{truc}(n)$.

$$T_{truc} = \frac{n}{BW} = \frac{5GByte}{570MByte/s} = \frac{5000MByte}{570MByte/s} \approx 8,77193s$$

We can now rearrange the compression formula with $n_c = \frac{n}{\frac{7}{2}} = \frac{5GB}{\frac{7}{2}} = \frac{10}{7}GB$:

$$\begin{aligned}
T_{trc}(n) &= T_{truc}(n) \\
T_{comp}(n) + T_{tr}(n_c) + T_{dec}(n_c) &= T_{tr}(n) && \mid -(T_{tr}(n_c) + T_{dec}(n_c)) \\
T_{comp}(n) &= T_{tr}(n) - T_{tr}(n_c) - T_{dec}(n_c) \\
\frac{n}{S_{comp}} &= T_{tr}(n) - T_{tr}(n_c) - T_{dec}(n_c) && \mid \cdot S_{comp} \\
n &= S_{comp} \cdot (T_{tr}(n) - T_{tr}(n_c) - T_{dec}(n_c)) && \mid \frac{1}{T_{tr}(n) - T_{tr}(n_c) - T_{dec}(n_c)} \\
S_{comp} &= \frac{n}{T_{tr}(n) - T_{tr}(n_c) - T_{dec}(n_c)} \\
S_{comp} &= \frac{n}{\frac{n}{BW} - \frac{n_c}{BW} - \frac{n_c}{S_{dec}}} \\
S_{comp} &= \frac{5000MByte}{\frac{5000MByte}{570MB/s} - \frac{\frac{10000}{7}MByte}{570MByte/s} - \frac{\frac{10000}{7}MByte}{2800MByte/s}} \\
&\approx 868,74028MByte/s
\end{aligned}$$

At least 868 MByte/s is required such that compressed data transfer is worth it.

2.

Without compression:

$$\begin{aligned}
T_{trucs}(n) &= \max(T_{tr}(n)) \\
&= \frac{n}{BW} \\
&= \frac{3000MByte}{400MByte/s} \\
&= 7,5s
\end{aligned}$$

With compression (and $n_c = \frac{n}{\frac{10}{3}} = \frac{3GByte}{\frac{10}{3}} = \frac{9}{10}GByte$)

$$T_{compress}(n) = \frac{3000MByte}{450MByte/s} = \frac{20}{3}s$$

$$T_{tr}(n_c) = \frac{n_c}{400MByte/s} = \frac{\frac{9000}{10}MByte}{400MByte/s} = 2,25s$$

$$T_{decompress}(n_c) = \frac{\frac{9000}{10} MByte}{1370 MByte/s} = \frac{90}{137} s \approx 0,65693 s$$

Therefore, $\max(T_{compress}(n), T_{tr}(n_c), T_{decompress}(n_c)) = \max(\frac{20}{3}s, 2,25s, \frac{90}{137}s) = \frac{20}{3}s$. This means that streamable compression is faster than no compression,

Problem 4

1.

With

$t.id$	$t.year < 2000$	round($t.rating$)	$t.title[0]$	$t.year$ mod 3
1	1	9	T	2
2	0	8	T	0
3	0	9	I	0
4	0	8	A	0
5	0	8	R	0
6	0	8	D	2
7	1	9	H	0
8	1	8	D	2
9	0	8	C	1
10	0	8	K	2

a) $R_1 = \{1, 7, 8\}$
$R_0 = \{2, 3, 4, 5, 6, 9\}$

b) $R_9 = \{1, 3, 7\}$
$R_8 = \{2, 4, 5, 6, 8, 9, 10\}$

c) $R_T = \{1, 2\}$
$R_I = \{3\}$
$R_A = \{4\}$
$R_R = \{5\}$
$R_D = \{6, 8\}$
$R_H = \{7\}$
$R_C = \{9\}$
$R_K = \{10\}$

d) $R_0 = \{2, 3, 4, 5, 7\}$
$R_1 = \{9\}$ $R_2 = \{1, 6, 8, 10\}$

2.

a) $\sigma_{year=2012}(T)$

Tree: 2015 $\rightarrow$ 2002 $\rightarrow$ 2008 $\rightarrow$ 2011 $\rightarrow$ 6 $\Rightarrow$ 4 indirections.

Hash: $2012 \mod 4 = 0$

$0 \rightarrow 8 \rightarrow 6$ 2 inderections.

b) $\sigma_{year=2017}(T)$

Tree: 2015 $\rightarrow$ 2020 $\rightarrow$ 9 2 indirections.

Hash: $2015 \mod 4 = 1$

$4 \rightarrow 9 \Rightarrow 1$ indirection.

c) $\sigma_{year=1999}(T)$

Tree: 2015 $\rightarrow$ 2002 $\rightarrow$ 1990 $\rightarrow$ 2000 $\rightarrow$ 1 $\Rightarrow$ 4 indirections. Movie 1 came out 1994, so nothing would be returned.

Hash: $1999 \mod 4 = 3$

$310 \rightarrow 5 \Rightarrow 2$ inderections.

d) $\sigma_{year=2001}(T)$

Tree: 2015 $\rightarrow$ 2002 $\rightarrow$ 1990 $\rightarrow$ 2000 $\rightarrow$ 4 $\Rightarrow$ 4 indirections.

Hash: $2001 \mod 4 = 1$

$1 \rightarrow 4 \Rightarrow 1$ inderection.

3.

It is clear that the hash index performs better than the tree index. Note that in our case, the number of elements is advantageous for the hash index, if more samples where to be added, the tree index will perform better. In our case, however, the hash index is the way to go.