Q1.

Storage Type	Capacity	Speed of Access	Cost
RAM	1GB – 4 GB	50ns – 60 ns	0.06 $/MB
Hard Disc	3 TB	3 ms	0.1 $/GB
Floppy Disc	1.44 MB	100 ms	0.18 $/MB
CD ROM	650 MB – 1400 MB	70 ms	0.24 $/GB
DVD	4.5GB – 16GB	110 ms	0.15 $/GB
Blu-Ray Optical Media	100GB	7 ms	0.12 $/GB
Magnetic Tapes	3TB	3-4 ms	0.4 $/GB

In terms of capacity blu-ray optical media become four times of its old capacity. Its capacity was 25 GB but its capacity is 100GB. Another significant increase is in magnetic tapes. Its capacity was 1 TB. It becomes three times of its old capacity. It becomes 3 TB.

In terms of cost magnetic tapes' cost become less than half of its old cost. It was 1$/GB but now it is 0.4$/GB

REFERENCE

- http://en.wikipedia.org/wiki/Hard_disk_drive#Data_transfer_rate
- http://en.wikipedia.org/wiki/History_of_hard_disk_drives
- http://www.unitarium.com/data
- http://en.wikipedia.org/wiki/Magnetic_tape#Data_storage
- http://www.imation.com/euc/pdfs/EUC_07_Qualls.pdf
- http://www.smartertechnology.com/c/a/Technology-For-Change/IBM-Brightens-Future-of-Magnetic-Tape-with-World-Record/
- http://en.wikipedia.org/wiki/Access_time
- http://www.blu-ray.com/faq/#bluray_speed
- http://www.cs.bilkent.edu.tr/~canf/CS351Fall2010/index.html (old homework)

Q2. Most common hard drives available in todays market place are the 5400 rpm ATA and 7200 rpm SATA drives.

a) For ATA hard drive, it has 5400 rpm. 5400 rpm means it has 5400 revolution per minute. Then we can calculate the time it takes for one full revolution with the following formula;

One full revolution = $\frac{60 x 1000}{5400}$ ≈ 11 msec

For SATA hard drive, it has 7200 rpm. 7200 rpm means it has 7200 revolution per minute. Then we can calculate the time it takes for one full revolution with the following formula;

One full revolution = $\frac{60 x 1000}{7200}$ ≈ 8 msec

b) Because of the head of the disk is equally likely to be at any byte position on the track after the arm has been moved, the average distance which the disk must rotate is half way around. Therefore, the average rotational delay is the time it takes the disk drive to make one half revolution. In other words, it is half of one full revolution. Therefore;

For ATA hard drive:
average rotational delay = $\frac{11}{2}$ = 5,5 msec

For SATA hard drive:
average rotational delay = $\frac{8}{2}$ = 4 msec

c) SATA drive has seek time s < 8,5 msec
ATA drive has seek time s < 12 msec

d) ATA drive has a maximum transfer rate of around 133 Mbs.
SATA drive has a typical transfer rate of around 150Mbs.

REFERENCE

- http://www.diy-computer-repairs.com/hard-drive-devices/
- http://www.tomshardware.com/forum/37236-32-seek-time-data-transfer-rate
- http://www.seagate.com/ww/v/index.jsp?vgnextoid=0732f141e7f43110VgnVCM100000f5ee0a0aRCRD

Q3.

T_s (Total Time for Sequential Processing)

Because of we have a sequential file at the beginning we have (s + r) to reach the beginning of the file. After that we read the records sequentially. We have b number of blocks and we read each block in ebt time. Therefore we have;

$T_s = s + r + (b \times ebt)$

T_r (Total Time for Random Processing)

Because of we are doing random processing the records are distributed randomly. Therefore for each accessing to the blocks we spent (s + r) time. Additionally, we read the block in btt time. Therefore for each block we spent (s + r + btt). Therefore we have;

$T_r = b \times (s + r + btt)$

For 100 blocks;

b = 100

$T_s = 16 + 8{,}3 + (100 \times 0{,}84) = 24{,}3 + 84 = 108{,}3$ msec

$T_r = 100 \times (16 + 8{,}3 + 0{,}8) = 2.510$ msec

For 1.000.000 blocks;

b = 1.000.000

$T_s = 16 + 8{,}3 + (1.000.000 \times 0{,}84) = 24{,}3 + 840.000 = 840.024{,}3$ msec

$T_r = 1.000.000 \times (16 + 8{,}3 + 0{,}8) = 25.100.000$ msec

For very large b (means b goes to ∞);

$\frac{Tr}{Ts} = \frac{b \times (s+r+btt)}{s+r+(b \times ebt)}$ because of b is very large we can ignore (s + r) while calculating T_s.

Therefore we have $\frac{Tr}{Ts} = \frac{b \times (s+r+btt)}{b \times ebt} = \frac{s+r+btt}{ebt} = \frac{16+8{,}3+0{,}8}{0{,}84} = 29{,}88 \approx 30$

Q4.

a) open input F_1

open output F_{12}

read F_1-Record from F_1

while(not EOF(F_1))

Search F_1-Read in F_2( )

if(not found**)**

write F_1-Read to F_{12}

read F_1-Record from F_1

end while

close F_1,F_{12}

<u>Search F_1-Record in F_2</u>

open input F_2

found = false

read F_2-Record from F_2

while (not found and not EOF(F_2))

found = F_1-Record == F_2-Record

read F_2-Record from F_2

end while

close F_2

return(found)

b) Firstly, let's calculate the time to read all the records in F_1 and denote it with Tall.

Tall = b x ebt = $\frac{n \times R}{B}$ x ebt

For common records we have the search time $\frac{Tall}{2}$ on average because on the average we read the half of the file.

For uncommon records we have search time **Tall** because we need to read all the file.

Now we are ready to derive a formula for the file processing time. Firstly, we need to read all the records in F_1 so we spent **Tall**. Then we know 70% of the records are common and reading common parts takes $\frac{Tall}{2}$ .Therefore we spent 0,7 x R x $\frac{Tall}{2}$. For the uncomman part with the same manner we spent 0,3 x R x **Tall** . After finishing reading the records we need to write the desired records to the output file. We are writing the uncommon part and we assume that writing time equals to reading time. Therefore to write the uncommon part we spent 0,3 x **Tall** . Then we have the file processing time:

$$T_{process} = (Tall) + (0{,}7 \times R \times \frac{Tall}{2}) + (0{,}3 \times R \times \mathbf{Tall}) + (0{,}3 \times \mathbf{Tall})$$

Let's calculate the file processing time.

n = 100.000 records and R = 800 bytes and B = 2400 bytes

$$Tall = b \times ebt = \frac{n \times R}{B} \times ebt = \frac{100.000 \times 800}{2400} \times 0.84 \text{ msec} = 28.000 \text{ msec} = 28 \text{ sec}$$

Then;

$$T_{process} = (28) + (0{,}7 \times 100.000 \times \frac{28}{2}) + (0{,}3 \times 100.000 \times 28) + (0{,}3 \times 28) = 1.820.000 \text{ sec}$$

$T_{process}$ = 30.333 minutes = 506 h.

c) Because of we read the F_1 record by record and we read all the records in F_1 we have n x (s + r)
Because of we read the F_2 record by record and we need to go to the beginning of the F_2 to search the records we have n x (s + r)
Because of we write F_{12} record by record and we write the uncommon part which is 30% of the all records we have 0,3 x n x (s + r)

In total we have n x (s + r) + n x (s + r) + 0,3 x n x (s + r)

= 2 x 100.000 x (24,3) + 0,3 x 100.000 x (24,3) = 5.589 sec

Q5. A)

a) Because of we have 10 MB of main memory is available to keep the records of F_1 we can read the F_1 in terms of blocks. In order to find uncommon records, first we read a block from F_1. For each record in this block we need to check the records in F_2. We need to do it record by record because we use all 10 MB of main memory for keeping the records of F_1. We continue to do this procedure until the blocks of F_1 finishes. While we are comparing the records, we are writing the uncommon records to the output file in terms of blocks.

- **b)** Using 10 MB of main memory affects seek time and rotational latency time in total. However, while calculating file processing time we ignore seek time and rotational latency time in total. Therefore the formula and the result are same with **Q4**.
- **c)** Because of we read F_1 in terms of 10 MB of blocks and we have 80MB/10MB = 8 blocks while reading the F_1 file we spent 8 x (s + r)
  Because of we use all the records in F_2 and there is no memory available for the file we spent n x (s + r)
  Because of we write in terms of blocks and we have 8 blocks to write
  we spent 8 x (s + r)

In total we spent 8 x (s + r) + 8 x (s + r) + n x (s + r) = 16 x (24,3) + 100.000 x (24,3)

= 2.430 sec

B) There will not be a significant change in file processing time. The reason is that using 20 MB of main memory to keep the records of F_1 decrease the amount of (s + r). However, during the calculation of the file processing time the effect of (s + r) can be ignored because they are relevantly small. Therefore, there will not be a significant change in file processing time.

Q6.

- **a)** Because of we can use 10 MB for both F_1 and F_2 we can read 5MB of F_1 and 5MB of F_2. In other words we can both read F_1 and F_2 in terms of blocks of 5MB. Therefore during the procedure, we read a 5MB block of F_1 and 5MB block of F_2. Then we compare them and find the uncommon records. We continue to the procedure until we finish the files. We also write the uncommon records to output file in terms of blocks.
- **b)** Using 10 MB of memory for both F_1 and F_2 decrease the amount of (s + r) in total. However, there is no change in terms of reading the block. In other words we still need to read 70% common and 30% uncommon records. The reading time will not change while we are using 10 MB of memory for both files. Therefore the formula and the calculations are same with **Q4**.
- **c)** Because of we read F_1 in terms of 5MB of blocks and we have 80MB/5MB = 16 blocks while reading the F_1 file we spent 16 x (s + r)
  Because of we use 5MB of blocks while reading the file F_2 and we have 80MB/5MB = 16 blocks to read we spent 16 x (s + r)
  Because of we write in terms of blocks and we have 80MB/5MB = 16 blocks to write we spent 16 x (s + r)

In total we have 3 x 16 x (s + r) = 48 x (24,3) = 1166 msec = 1,2 sec