

# Homework 1

---

## Problem 1 (Disks, 70 Points (10 each))

Given a disk with the following configurations:

- There are 4 (double-sided) platters, each surface has 4096 tracks, 512 sectors per track.
- A sector holds 1024 Bytes.
- Disk rotates at 7200 rpm.
- Seek time: A warm-up time is 2ms, and then the arm movement covers 400 tracks per ms.
- Reading one sector takes 0.08 ms.
- A disk block is of size 8K Bytes (8 \* 1024 bytes).

### 1.1 Compute the disk capacity in GBs

```
Disk capacity (bytes)
= 2 sides/platter * 4 platters/disks * 4096 tracks/side * 512 sectors/track
* 1024 bytes/sector
= 17,179,869,184 bytes
```

### 1.2 Compute the disk capacity in terms of the number of blocks it can hold

```
Disk capacity (blocks)
= 17,179,869,184 bytes / (8 * 1024) bytes/block
= 2,097,152 blocks
```

### 1.3 What is the minimum, maximum, and average time needed to read one block from the disk? You need to divide the time into its three components (seek, rotation latency, transfer)

Minimum time (best case): head is already on block that we want → only transfer time

```
Minimum time
= 0.08 ms/sector * 8 sectors/block
= 0.64 ms
```

Maximum time (worst case): head is on the opposite side of the disk, and the block we want is on the opposite side of the disk. And the block we want just passed the head, so we need to wait for a full rotation → seek time + rotation latency + transfer time.

```
Maximum time
= seek time (warm-up + arm movement) + rotation time + transfer time
= 2 ms + 4096 tracks / 400 tracks/ms + 1000 / (7200 rpm / 60) + 0.08 ms * 8
```

```
sectors/block
= 2 ms + 10.24 ms + 8.33 ms + 0.64 ms
= 21.21 ms
```

Average time (average case): block we want is 1/2 of the total number of tracks away and 1/2 revolution away → seek time + rotation latency + transfer time

```
Average time
= seek time (warm up + arm movement) + rotation time + transfer time
= 2 ms + 4096 tracks / 2 / 400 tracks/ms + (1 / 2) * 1000 / (7200 rpm / 60)
+ 0.08 ms/sector * 8 sectors/block
= 2 ms + 5.12 ms + 4.17 ms + 0.64 ms
= 11.93 ms
```

1.4 Given a file that consists of 200,000 records, and each record is 256 bytes. No record is allowed to span multiple blocks

- How many records fit in one block?

```
Records per block
= 8 * 1024 bytes / 256 bytes
= 32 records
```

- How many blocks are needed to store the file?

$200,000 \text{ records} / 32 \text{ records/block} = 6,250 \text{ blocks}$

- How many sectors are needed to store the file?

$6,250 \text{ blocks} * 8 \text{ sectors/block} = 50,000 \text{ sectors}$

1.5 Assume Blocks B1, B2, B3, ..., B15, are stored on the same track (track number 200) and on adjacent sectors. If the disk head is positioned at track 0, what is the I/O time (seek + latency + transfer) needed to read these 15 blocks? (Assume a half rotation is needed to reach the first block of these 15 blocks)

```
I/O time
= warm-up + arm movement + rotation time + transfer time
= 2 ms + 200 tracks / 400 tracks/ms + (1 / 2) * 1000 / (7200 rpm / 60) + 15
* 8 sectors/block * 0.08 ms/sector
= 2 ms + 0.5 ms + 4.16 ms + 9.6 ms
= 16.26 ms
```

1.6 To optimize the reading of the file, we may store adjacent blocks, e.g., B1, B2, B3, ..., on the same cylinder. Given the disk configuration above, how many blocks can be aligned on

the cylinder position to be read by the disk arms at the same time?

```
Blocks per cylinder
= 8 tracks/cylinder * (512 sectors/track / 8 sectors/block)
= 512 blocks
```

1.7 Referring to Q4 and the number of blocks you calculated, i.e., B1, B2, B3, ... until the last block Blast. Describe the best way to store these blocks on disk to speed up the sequential read (reading the entire file block by block). Then measure out the average time needed to read the file given this organization. Hint: Make use of cylinders

For optimal sequential read of the entire file, its blocks should be stored contiguously in a cylinder order. This way, the disk head can read in the entire 512 blocks in a cylinder without any seek time (only one full rotation latency + transfer time). Afterwards, it can seek to the adjacent cylinder with minimal seek time to read in the next 512 blocks.

```
Cylinders used
= 6,250 blocks / 512 blocks/cylinder
= 12.21 cylinders
```

```
Remaining blocks
= 6,250 blocks - 12 * 512 blocks
= 106 blocks
```

```
Time to read each cylinder: since head has to be idle for a full rotation,
we have to warm it up
again before seeking to the next cylinder
= seek time (warm up + arm movement) + rotation time + transfer time
= 2 ms + 1 track / 400 tracks/ms + 1000 / (7200 rpm / 60) + 512 * 8
sectors/block * 0.08 ms/sector
= 2 ms + 0.0025 ms + 8.33 ms + 327.68 ms
= 338.01 ms
```

```
Time to read remaining blocks
= 2 ms + 1 track / 400 tracks/ms + 1000 / (7200 rpm / 60) + 106 * 8
sectors/block * 0.08 ms/sector
= 2 ms + 0.0025 ms + 8.33 ms + 67.84 ms
= 78.17 ms
```

Assume that the head's initial position is 1/2 rotation and 1/2 of the total number of tracks away from the first block of the first cylinder.

```
Time to read the entire file
= 4096 tracks / 2 / 400 tracks/ms + (1 / 2) * 1000 / (7200 rpm / 60) - 2 ms
- 0.0025 ms + 12 * 338.01 ms + 78.17 ms
= 5.12 ms + 4.17 ms - 2 ms - 0.0025 ms + 12 * 338.01 ms + 78.17 ms
= 4141.58 ms
```

## Problem 2 (Record Organization, 30 Points (10 each))

Assume a database table with the following fields

- ID (8 bytes), Name (30 bytes), age (4 bytes), DoB (12 bytes), gender (1 Byte), Address (60 bytes), state (4 bytes).
- Each record on disk has a header part of size 10 bytes.

2.1 What will be the record size if each field has to start at 4-byte boundaries?

```
Record header size:
- 10 bytes data + 2 bytes padding
Record data size:
- ID: 8 bytes data
- Name: 30 bytes data + 2 bytes padding
- Age: 4 bytes data
- DOB: 12 bytes
- Gender: 1 byte data + 3 bytes padding
- Address: 60 bytes data
- State: 4 bytes data

Total = 10 + 2 + 8 + 30 + 2 + 4 + 12 + 1 + 3 + 60 + 4
      = 136 bytes
```

2.2 What will be the record size if each field has to start at 8-byte boundaries?

```
Record header size:
- 10 bytes data + 6 bytes padding
Record data size:
- ID: 8 bytes data
- Name: 30 bytes data + 2 bytes padding
- Age: 4 bytes data + 4 bytes padding
- DOB: 12 bytes data + 4 bytes padding
- Gender: 1 byte data + 7 bytes padding
- Address: 60 bytes data + 4 bytes padding
- State: 4 bytes data + 4 bytes padding

Total = 10 + 6 + 8 + 30 + 2 + 4 + 4 + 12 + 4 + 1 + 7 + 60 + 4 + 4 + 4
      = 160 bytes
```

2.3 Assuming a block of size 4K bytes, the block uses 80 bytes of its own header (block header). How many records can fit in one block? Report the number under the 4- and 8-bytes boundaries

```
Remaining space for records = 4096 bytes - 80 bytes = 4016 bytes

4-byte boundary: Records per block = 4016 bytes / 136 bytes/record = 29.53
```

records

So, 29 records can fit in one block under 4-byte boundaries.

8-byte boundary: Records per block =  $4016 \text{ bytes} / 160 \text{ bytes/record} = 25.1$  records

So, 25 records can fit in one block under 8-byte boundaries.