CS143: Homework #5 (Disk and File)

We want to store the table created by the following SQL statement into a disk.

```
CREATE TABLE Class(
  dept  CHAR(2),
  cnum  INTEGER,
  sec   INTEGER,
  unit  INTEGER,
  year  INTEGER,
  quarter INTEGER,
  title CHAR(30),
  instructor CHAR(20)
)
```

We need to store tuples for 1,000 classes that have been offered so far. 10 classes are offered every year. The tuples are stored in random order (i.e., they are not sequenced by any attribute). A disk of the following parameters is used for storing the table.

- 3 platters (6 surfaces)

- 10,000 cylinders

- 500 sectors per track

- 1024 bytes per sector

- 6,000 RPM rotational speed

- 10ms average seek time

1. What is the average time to read a random sector from the disk?

   ANSWER:
   (seek time) + (rotational delay) + (transfer time) = 10ms + 5ms + 0.02ms = 15.02ms

2. Assume one disk block corresponds to one disk sector. How many disk blocks are needed to store the above table with 1,000 tuples?

   ANSWER:
   72 blocks. $\lfloor\frac{1024 \text{ bytes/block}}{1 \text{ tuple/72 bytes}}\rfloor = 14$ tuples/block. $\lceil\frac{1000 \text{ tuples/table}}{14 \text{ tuples/block}}\rceil = 72$ blocks/table.

3. We want to run the following query by scanning the entire table.

   ```
   SELECT * FROM Class WHERE year = 2005
   ```

Assuming that all blocks for the table is allocated sequentially, how long will it take to run the query? Assume that the disk head is not on the same track where the first block of the table is stored.

ANSWER:
```
(seek time) + (rotational delay) + (transfer time) = 10ms + 5ms + 72*0.02ms = 16.44ms
```

4. Now assume that due to frequent updates to the table, disk blocks are allocated such that, on average, sequentiality is broken every three blocks. That is, the table is stored in 24 randomly located "clusters" of 3 consecutive blocks. Assuming that we scan the entire table to execute the above query, how long will it take?

ANSWER:
```
24 * ((seek time) + (rotational delay) + (transfer time)) = 24 * (10ms + 5ms + 3*0.02ms)
= 361.44ms
```

5. Now assume that we have a non-clustering index on the year attribute and the index has already been loaded into main memory. None of the disk blocks containing the Class table has been cached in main memory. What is the expected time to run the above query? Is it helpful to create a nonclustering index to run this query?

ANSWER:
```
150.2ms.  Since the tuples are not clustered by the search key, we will need to do 10
random IOs to retrieve all 10 tuples.  Therefore, 10*(10ms+5ms+0.02ms) = 150.2ms.  When
tuples are not stored sequentially, using the index may actually slow down the query execution.
```