External Sorting Questions Solutions

1. Suppose we have a disk with the following specifications: Speed is 8200RPM, average seek time is 7.9ms, and track to track seek delay is 1.8ms. The rotational delay is then

   a) About 14.634ms
   b) About 7.9ms
   c) About 7.31ms
   d) About 3.66ms

Rotational delay = time taken to do half a rotation = ½*((60 sec/minute)/8200)=3.66ms

2. When using the Simple Merge Sort algorithm, the size of the output run

   a) Is fixed
   b) Varies randomly after each step depending on the order of the records in the original file
   c) Is doubled in each step

The answer is c). The size of the run is doubled after each step.

3. The size of the following disk allocation unit is determined by the Operating System

   a) A byte
   b) A sector
   c) A cluster
   d) A track

As explained in the class and in the book, it is the cluster.

4. We have a 1KB file of records that we want to sort, where each record is 16 bytes. Suppose the records are arranged in the file such that the max is in the middle and the key values go down until reaching the minimums at both ends. Here is an example:

Top of file`...,13,27,32,43,54,58,`**`60`**`,55,52,41,30,23,...`Bottom of file

The Replacement Selection algorithm, with a min heap of 4 nodes, will produce

   a) 8 runs
   b) 9 runs
   c) 10 runs
   d) 16 runs
   e) 32 runs

Note that the record keys increase until half way into the file. Remember as the minheap is fed with records whose keys are larder than those of the ones that were sent to the output buffer, then the size of the heap does not change. As a result, the outputted records go into the same output run. So up to the middle of the file, we have one output run. Now, past the middle and as the keys keep on decreasing, the size of the heap starts shrinking.

To figure out the number of runs:

½*1024/16=32 records. These 32 records will fill the heap 32/4=8 times. Because the keys arrive to the heap in decreasing order, the size of the heap decreases at each step and hence we will get 8 output runs.
So the total number of runs = 1 (containing half of the total records) + 8 (each having 4 records) = 9

5. Suppose we have the following two disks:
- Disk 1: 8200RPM, average seek time is 9.5ms, track to track seek time is 1.5ms
- Disk 2: 5600RPM, average seek time is 5.2ms, track to track seek time is 1.75ms

Given the same operating system,
a) We will always read files faster from disk 1
b) We will always read files faster from disk 2
c) None of the above

Depending on the status of the disk; badly fragmented (worst case) on one end versus totally available disk (best case), the same file will either take longer or faster to read from disk 1 when compared to disk 2. To convince yourself, take a file of, say, 1 MB in size and try it under both conditions.
Therefore, the answer is c).

6. Secondary memory refers to
a) Cache memory
b) Random Access Memory
c) Hard Drives
d) Removable Disks, i.e. Floppies, Compact Disks, etc.

Secondary memory always refers to Hard Disks (mentioned in class more than 1 time)

7. If we have 4800 records of employees in a file, where each record consists of a first name (16 characters), a last name (16 characters), an ID (long integer), and a salary (double). Suppose on our computer system, a character is 1 byte, a long integer is 8 bytes, and a double is also 8 bytes. The hard drive on our system has a sector of 520 bytes, the cluster size is 6KB, and the track is 2048 sectors. Our file will then occupy
a) 225KB
b) 228KB
c) 222KB

Size of each record = 16*1+16*1+1*8+1*8=48 bytes and hence our file is 4800*48=230400=225KB. To know how many clusters it occupies, divide 225 by the cluster size to get 225/6=37.5, which means that our file occupies 38 clusters.
Then the actual size of the file (what it occupies on the disk) = 38*6=228KB

8. The external sort algorithms were developed

a) To sort records in files, in general
b) To sort records in multiple files
c) To sort records in huge files
d) To sort data faster

This was mentioned many, many times. It is to sort files that do not fit in memory: huge files.

9. If we have three sorted buffers of records that we want to sort through the Multiway Merge algorithm, and the first buffer contains 300 records, the second one has 350 records, while the third has 250 records. To do the sorting, the algorithm will take

a) 250 steps
b) 300 steps
c) 350 steps
d) 900 steps

Remember that the Multiway Merge algorithm always inspects the first records in all the run files and selects the **one** with minimum key and sends it to the output file. Hence, it is sending one record at a time. Well, we have 300+350+250=900 records and the answer becomes d).

10. Suppose we have a buffer whose size is 32 bytes and uses the **least recently used (LRU)** algorithm. If the file we trying to read has 50 sorted integers (an integer is 4 bytes) in the range 1 to 50. Suppose the following series of values are read from the file:
5, 2, 5, 12, 3, 12, 5, 9, 3, 2, 4, 1, 5, 9, 4, 1, 15, 3, 7, 2, 4, 9, 12, 8, 1, 8, 6
Then the content of the buffer (order doesn't matter) after the last file read is

a) 1 2 3 4 5 6 9 12
b) 1 2 3 4 5 6 7 8
c) 1 2 3 4 6 8 9 12
d) 1 2 4 6 7 8 9 12
e) 2 3 4 5 6 7 8 9

Using an LRU, every time a value is accessed in the buffer, it gets moved to the front within the buffer to indicate the order of its access (buffer is initially empty):

new val	Buffer Contents; the position of values within the buffer indicate the order of their access							
5	5							
2	2	5						
5	5	2						
12	12	5	2					
3	3	12	5	2				
12	12	3	5	2				
5	5	12	3	2				
9	9	5	12	3	2			

3	3	9	5	12	2			
2	2	3	9	5	12			
4	4	2	3	9	5	12		
1	1	4	2	3	9	5	12	
5	5	1	4	2	3	9	12	
9	9	5	1	4	2	3	12	
4	4	9	5	1	2	3	12	
1	1	4	9	5	2	3	12	
15	15	1	4	9	5	2	3	12
3	3	15	1	4	9	5	2	12
7	7	3	15	1	4	9	5	2
2	2	7	3	15	1	4	9	5
4	4	2	7	3	15	1	9	5
9	9	4	2	7	3	15	1	5
12	12	9	4	2	7	3	15	1
8	8	12	9	4	2	7	3	15
1	1	8	12	9	4	2	7	3
8	8	1	12	9	4	2	7	3
6	**6**	**8**	**1**	**12**	**9**	**4**	**2**	**7**

The answer is d).