You will submit your solution to this assignment to the Curator System (as `HW1`). Your solution must be either a plain text file (e.g., NotePad) or a MS Word document; submissions in other formats will not be graded.

Partial credit will only be given if you show relevant work.

---

1. [20 points] Using any relevant theorems from the notes, conjecture a simple function g such that f is $\Theta(g)$, and prove that your conjecture is correct if:

$$f(n) = n^2 \log n + n^3 + 1000$$

Since n is a strict upper bound for log n, from Theorem 5, it seems reasonable that n^3 is a strict upper bound for n^2 log n. However, Theorem 5 doesn't settle the issue, so you need to apply the limit theorem:

$$\lim_{n\to\infty} \frac{n^2 \log n + n^3 + 1000}{n^3} = \lim_{n\to\infty}\left(\frac{\log n}{n} + 1 + \frac{1000}{n}\right) = \lim_{n\to\infty}\left(\frac{\log n}{n}\right) + 1 + 0 = \lim_{n\to\infty}\left(\frac{1/n\ln 2}{1}\right) + 1$$

$$= \lim_{n\to\infty} \frac{1}{n\ln 2} + 1 = 0 + 1 = 1$$

Therefore, by the limit theorem, f is $\Theta(n^3)$.

---

2. [20 points] Let α be an arbitrary positive constant, and define two functions:

$$f(n) = \log n \quad \text{and} \quad g(n) = n^\alpha$$

Using any theorems from the notes, prove that f is O(g) but f is not Θ(g).

Again, apply the limit theorem:

$$\lim_{n\to\infty} \frac{\log n}{n^\alpha} = \lim_{n\to\infty} \frac{1/n\ln 2}{\alpha n^{\alpha-1}} = \lim_{n\to\infty} \frac{1}{\alpha n^\alpha \ln 2} = 0$$

So by the limit theorem, f is $O(n^\alpha)$ but not $\Theta(n^\alpha)$.

3. [20 points] Suppose that f and g are non-negative functions such that f is Θ(g). Is it necessarily true that:

$$2^{f(n)} \text{ is } \Theta\left(2^{g(n)}\right)$$

If so, prove it. (You may assume that the limit referred to in Theorem 8 exists.) If no, give a specific counter-example and show that it is a counter-example.

The assumption the limit exists does not result in a proof, since:

$$\operatorname*{limit}_{n\to\infty} \frac{2^{f(n)}}{2^{g(n)}} = \operatorname*{limit}_{n\to\infty} 2^{f(n)-g(n)}$$

The assumption that f is Θ(g) may seem to tell us anything useful here, since it would at most imply something about the limit of the ratio of f to g, not their difference.

That observation, though, suggests that there may be a simple counter-example. Does the fact that the limit of f/g is finite really imply that the limit of f-g is finite? Consider a simple example, say:

$$f(n) = n \text{ and } g(n) = 2n$$

Now, f(n)/g(n) = 2, so the limit of the ratio is clearly 2 and so f is clearly Θ(g). But, on the other hand, f(n) – g(n) is n, and the limit of n isn't finite. That is:

$$\operatorname*{limit}_{n\to\infty} \frac{2^{n}}{2^{2n}} = \operatorname*{limit}_{n\to\infty} \frac{1}{2^{n}} = 0$$

and so 2^f is clearly not $\Theta(2^g)$.

---

4. [20 pts] Assume a system uses a hard drive with the following physical characteristics:

total capacity	128 GB
# of platters	8
# of tracks per surface	16384
# of sectors per track	2048
cluster size	4 KB
spindle speed	10000 RPM
head start time	1 ms
track to track seek time	0.01 ms

In answering the following questions, express all final time values to the nearest hundredth of a millisecond (8.33 ms).

a) What is the average random head seek time for this drive?

From the notes, the average random seek time is: (1/3)*(#tracks per surface)*(track to track seek time) + (head start time)
Using the data above, that works out to be 55.61 ms (which is not exactly great performance).

b) What is the average rotational latency for this drive?

From the notes, the average rotational latency is the time required for ½ of a full rotation: $\frac{1}{2} \times \frac{1000}{R}$

where R is the number of rotations per second. Since the spindle speed is 10000 RPM, R is 10000/60 and hence the rotational latency is

$$\frac{1}{2} \times \frac{1000}{\frac{10000}{60}} = 3 \text{ ms}$$

c) What is the average total time required to read one randomly-chosen sector from this drive?

The time is the average seek time plus the average rotational latency plus the time for a sector to rotate past the read head (the transfer time). The first two were determined above. From the notes, the transfer time is given by

$$\text{Transfer}(n) = \left(\frac{n}{S_T}\right) \times \left(\frac{1000}{R}\right)$$

where n is the number of sectors to be read and S_T is the number of sectors per track and R is as in part b. Using the given values, the transfer time is

$$\left(\frac{1}{2048}\right) \times \left(\frac{1000}{\frac{10000}{60}}\right) = \frac{6}{2048} \approx 0.0029 \text{ ms}$$

So, the average time to read a random sector is about 55.61 + 3 + .0029 ms or 58.61 ms.

d) What is the average total time required to read a file of 10 MB from this drive if the clusters are randomly scattered on the drive?

The answer would be (# of clusters file occupies) * (average time to read one cluster). Since the cluster size is 4 KB and the file is 10 MB, the file will occupy exactly 2650 clusters.

The time to read one cluster would be the sum of the average seek time plus the average rotational latency plus the transfer time for a single cluster. We already know the first two values. To find the transfer time we need to know how many clusters there are on a track, or how many sectors are in a cluster. Neither of those is given, but either can be calculated from the given drive parameters. The sector size is given by:

$$\frac{\text{\#bytes on disk}}{\text{\#surfaces}} = \text{\#bytes per surface} = \frac{128 \times 2^{30}}{16} = 128 \times 2^{26}$$

$$\frac{\text{\#bytes per surface}}{\text{\#tracks per surface}} = \text{\#bytes per track} = \frac{128 \times 2^{26}}{16384} = \frac{128 \times 2^{26}}{2^{14}} = 128 \times 2^{12} = 2^{19}$$

$$\frac{\text{\#bytes per track}}{\text{\#sectors per track}} = \text{\#bytes per sector} = \frac{2^{19}}{2048} = \frac{2^{19}}{2^{11}} = 2^{8}$$

Since a cluster is 4 KB (2^{12} bytes), each cluster contains 16 sectors. Thus, the transfer time is given by:

$$\left(\frac{16}{2048}\right)\times\left(\frac{1000}{\frac{10000}{60}}\right)=\frac{96}{2048}\approx 0.0468759\text{ ms}\approx 0.05\text{ ms}$$

The average time to read one random cluster is 55.61 + 3 + .05 = 58.66 ms. Since the file contains 2560 clusters, the total read time is about 2560 * 58.66 or 150169.6 ms (which is about 2.5 minutes).

5. [20 points] Consider solving a problem using an algorithm whose complexity is $\Theta(N^2)$. Estimate the running time of the algorithm if:

a) $N = 2^{12}$ (4096) and the hardware is capable of executing 2^{24} instructions per second.

Using the complexity bound, the number of instructions to be executed can be estimated to be:

$$\#\text{instructions} = \left(2^{12}\right)^2 = 2^{24}$$

So, the time to execute the algorithm can be estimated to be:

$$\text{time} = \frac{\#\text{instructions}}{\text{instructions per second}} = \frac{2^{24}}{2^{24}} = 1\text{ second}$$

b) $N = 2^{16}$ (65536) and the hardware is capable of executing 2^{24} instructions per second.

This takes the same logic, but a different value of N, so:

$$\text{time} = \frac{\left(2^{16}\right)^2}{2^{24}} = \frac{2^{32}}{2^{24}} = 2^8 = 256 = 4\text{ minutes }16\text{ seconds}$$

Express your answers in days, hours, minutes and seconds. (Not in total seconds unless the time is shorter than 1 minute.)