CS280 – Data Structures – Quiz

DigiPen login: _____

1. True or false that function on the left belongs to a big-O class on the right:

$$\begin{aligned} n^3 - 10n + \log n &\in O(n^4) \\ n^3 - 10n + \log n &\in O(n^3) \\ n^3 - 10n + \log n &\in O(n^2) \\ \log n &\in O(\sqrt{n}) \\ \sqrt{\log n} &\in O(\sqrt{n}) \\ \left(\log n^2\right) &\in O(\sqrt{n}) \\ \left(\frac{3}{2}\right)^n &\in O(n^2) \end{aligned}$$

2. Use only

   - $O(1)$,
   - $O(n^k \log^m n)$ (provide k and m) – examples: $O(\sqrt{n})$, $O(\log^2 n)$, $O(n \log n)$, $O(n^3)$, ...
   - $O(b^n)$ (provide b) – examples: $O(2^n)$, $O((1.2)^n)$, ...

   For each of the next functions provide big-O notation:

   - $f(n) = 1 + n + n^2$
   - $f(n) = \log n + n + \sqrt{n}$
   - $f(n) = \log^5 n + n + \sqrt{n}$
   - $f(n) = \log^5 n + \log^2 n$
   - $f(n) = n \log n + \log^2 n$
   - $f(n) = n \log n + n^2 \log n$
   - $f(n) = n^2 \log n + n^2 + n$
   - $f(n) = n^2 \log^2 n + 2^n$
   - $f(n) = 3^n + 2^n$

3. You are given an $O(n^2)$ implementation of an algorithm. You ran it on an input of size 100 and the runtime was 1 second. How long will it take to execute on an input of size 300?

4. You are given an implementation of an algorithm which takes exactly n^2 ticks to execute on a input of size n and another implementation of the same algorithm which is $9n + 10$ ticks.

   - Which implementation will you use on an input of size 2?
   - Which implementation will you use on an input of size 20?
   - At what value of n will you switch from one implementation to another?

5. What are the best and worst run-times of the algorithm below (use big-O notation, provide details):

```
int a[N] = { .... };
int i = N-1;
while ( i >= 0 ) {
    cout << a[ i ] << " ";
    --i;
}
```

6. What are the best and worst run-times of the algorithm below (use big-O notation, provide details):

```
int a[N] = { .... };
int size = N;
int i = 0;
while ( size > 0 ) {
    if ( a[i] < i ) {
        i += size/2;
    }
    cout << a[ i ] << " ";
    size /= 2;
}
```

7. What are the best and worst run-times of the algorithm below (use big-O notation, provide details):

```
int a[N][N] = { .... };
for ( int i=0; i<N-2; ++i ) {
    for ( int j=i; j<N; j += 2 ) {
        cout << a[i][j] << " ";
    }
}
```

8. What are the best and worst run-times of the algorithm below (same as above, just 2 changed to 10) (use big-O notation, provide details):

```
int a[N][N] = { .... };
for ( int i=0; i<N-10; ++i ) {
    for ( int j=i; j<N; j += 10 ) {
        cout << a[i][j] << " ";
    }
}
```