Student Name: Nikita Semeniuk
Student ID: 2722726

Assignment 1: Asymptotic complexity

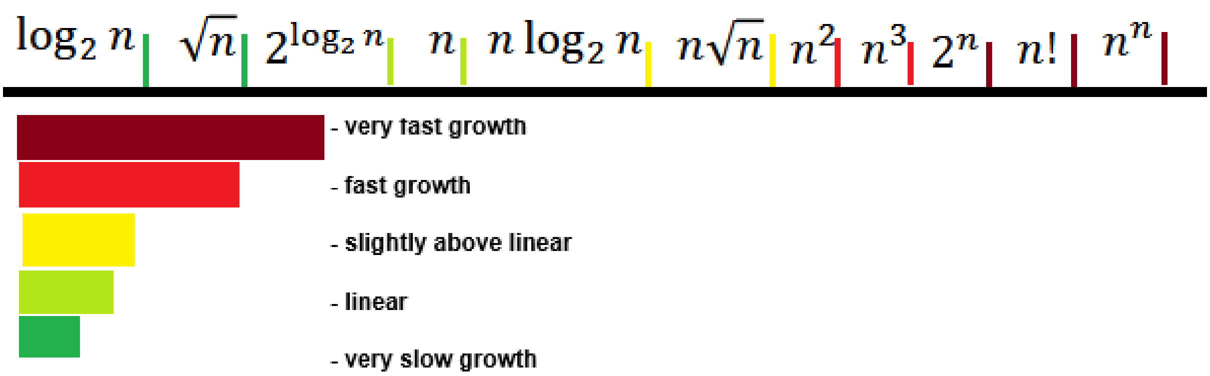
Rank the following terms in ascending order of asymptotic complexity and explain your solution:

$n! \quad n^2 \quad n\log_2 n \quad n^3 \quad 2^n \quad n^n \quad \sqrt{n} \quad 2^{\log_2 n} \quad \log_2 n \quad n\sqrt{n} \quad n$

For establishing the correct order it helped to set my baseline at n as a linear case and go from there (better performing < n < worse performing). From here on out:

- **n! -** starting with n! - factorials tend to blow up fast in value. Even small values of n clearly show this trend: !3 = 6, !6 = 720, !8 = 40320. So definitely on the right side.
- **n^2 -** quadratic, not so bad when the values of n are not too big. So while worse than linear **n**, still not nearly as bad as **n!**.
- **n*log2(n) -** this is linear **n** multiplied by a small factor, so slightly above linear time **n**.
- **n^3 -** just above **n^2** since it has higher power.
- **2^n -** exponential growth, grows really fast but not as bad as factorial. For comparison: When n = 5: 2^5 = 32, 5! = 120.
- **n^n -** very bad case. Starts growing faster than a factorial even at low values of n. When n = 2: 2^2 = 4, 2! = 2. With larger values, behaves even worse: n = 5 (5^5 = 3125 vs 5! = 120).
- **sqrt(n) -** slightly less than **n**.
- **2^(log2(n)) -** according to the logarithm property this is just a regular **n** in disguise.
- **log2(n) -** the slowest growth so far, slightly slower than **sqrt(n)**.
- **n*sqrt(n) -** faster growth than **n*log2(n)**, but still lower than quadratic and exponential factors.
- **n -** linear time.

This leads to the following order:



Testing with **n = 12** further confirms these observations:

Function	**Result**
log2(n)	~3.58
sqrt(n)	~3.46
2^(log2(n))	12
n	12
n*log2(n)	~43.02
n*sqrt(n)	~41.57
n^2	144
n^3	1,728
2^n	4,096
n!	479,001,600
n^n	8,916,100,448,256