Solutions to Ex 2

1. **Result 1** $p(x) = a_m x^m + \dots + a_0$, $a_m > 0$.

Put $K = \max(|a_0|, \dots, |a_m|)$. Then for $x \geq 1$, we have that

$$\begin{aligned} a_m x^m + \dots + a_0 &\leq |a_m| x^m + \dots + |a_0| \\ &\leq K(x^m + \dots + 1) \leq K(x^m + \dots + x^m) \\ &= \underbrace{K(m+1)}_{\text{Const.}} x^m \end{aligned}$$

Thus $p(x) = O(x^m)$.

Result 2. $n > m > 0$. $\frac{x^m}{x^n} = \frac{1}{x^{n-m}} \to 0$ as $x \to \infty$.

It follows that $x^m \leq x^n$ for x big enough. Thus $x^m = O(x^n)$.

Result 3 If $a > 1$ then $x^m = O(a^x)$. Convert this to an exponential

$a = e^{\log_e(a)}$. $a^x = (e^{\log_e(a)})^x = e^{bx}$ where $b = \log_e(a)$.

$$e^{bx} = 1 + (bx) + \frac{1}{2!}(bx)^2 + \frac{1}{3!}(bx)^3 + \dots$$

$$\therefore \frac{m!}{b^m} e^{bx} = (\text{stuff} \geq 0) + x^m + (\text{stuff} \geq 0)$$

Thus $x^m \leq \frac{m!}{b^m} e^{bx}$. It follows that $x^m = O(e^{bx}) = O(a^x)$.

Result 4 $\log(x) = O(x^\alpha)$. Put $y = \log(x)$. Then $x = e^y$.

By our result above $y = O((e^\alpha)^y)$ and so $\log(x) = O(x^\alpha)$.

---

2. Functions arranged in order from very fast to very slow

$\log\log n$, $\sqrt{\log n}$, $\log n$, $\log(n^2+1)$, $(\log n)^2$, $n \log n \log\log n$

$n^{3/2}$, n^2, $n^2 \log n$, 2^n, 3^n, $n!$, n^n, 2^{n^2}

Justifications

$\log\log n = O(\sqrt{\log n})$

By Result 4, $\log x < x^{1/2}$ for x large enough. Can replace x by $\log n$ for n large enough.

$\sqrt{\log n} = O(\log n)$

By Result 2, $x^{1/2} < x$ for x large enough. Thus $(\log n)^{1/2} < \log n$ for n large enough.

$\log n = O(\log(n^2+1))$

$n < n^2+1$ for n large enough. Thus $\log n < \log(n^2+1)$.
We also use the fact that $\boxed{x < y \Rightarrow \log x < \log y}$

$\log(n^2+1) = O((\log n)^2)$

$n^2+1 \le n^2+n^2 \le 2n^2$ for n large enough.
Thus $\log(n^2+1) \le \log(2n^2) = \log 2 + 2\log n$ using
$\boxed{\log(xy) = \log(x) + \log(y)}$ But $\log 2 < \log n$ for n large enough.
Hence $\log(n^2+1) \le 3\log n$ for n large enough.
By Result 2, $x < x^2$ for x large enough. Thus $\log n < (\log n)^2$.
Hence $\log(n^2+1) \le 3(\log n)^2$ for n large enough.

$(\log n)^2 = O(n \log n \log\log n)$

By Result 2, $\log n < n$ for n large enough.
Thus $(\log n)^2 < n\log n$ for n large enough.
Now $\log\log n > 1$ when n big enough. Hence
$(\log n)^2 < n\log n\log\log n$ for n large enough.

3 (a). Let the current speed of executing one step be τ hours. Then $1 = N^5 \tau$. If the new computer is 10,000 times faster then $1 = \bar{N}^5 \frac{\tau}{10{,}000}$ where $\bar{N}$ is the new max. input length.

We get $\bar{N} = N \sqrt[5]{10^4}$

$\approx \underline{\underline{6.31N}}$

(b) As above, $1 = 2^N \tau$ and $1 = 2^{\bar{N}} \frac{\tau}{10000}$.

Thus $2^{\bar{N}} = 2^N \cdot 10^4$. It follows that

$$\bar{N} = N + \log_2(10^4) \simeq \underline{\underline{N + 13.28}}$$

4. A and B are two $n \times n$ matrices.

$$(AB)_{ij} = i\text{th row of } A \cdot j\text{th column of } B$$
$$= \sum_{k=1}^{n} a_{ik} b_{kj} \quad \text{which consists of } n \text{ multiplications.}$$

This has to be repeated for all n^2 entries of AB. It follows that n^3 multiplications need to be carried out.

5. The key point is that the list is ordered. Let the list have n entries. Compare input with middle of the list $\lfloor n/2 \rfloor$: it is either in the first half or the second half. Now repeat with whichever is appropriate. The no. of comparisons is $1 + \lfloor \log_2 n \rfloor$ - the no. of binary digits in the base 2 rep of n.

[Maybe better: label the entries 1 to n in binary in increasing order. In each comparison, we are finding the binary digit of the item we are looking for, one digit at a time from the left.]

6. $31 = \lfloor \sqrt{997} \rfloor$. This is prime.

7. $347 \times 27{,}953$ (how did you do it?)