National University of Singapore
School of Computing
CS1101S: Programming Methodology
Semester I, 2024/2025

S3-extra
Extra Material

If there is time after working on the in-class sheet, feel free to use this extra material, following textbook section 1.2.4.

Problems:

1. Consider the following function:

   ```
   function expt(b, n) {
       return n === 0 ? 1 : b * expt(b, n-1);
   }
   ```

   Does ‘expt‘ give rise to an iterative or recursive process? Use the Θ notation to characterize the time and space consumption of `expt` as the argument `n` grows.

   Use the Θ notation to characterize the time and space consumption of `expt` as the argument `b` grows.

2. Consider following relationship

   $$b^n = \begin{cases} (b^{n/2})^2 & \text{if } n \text{ is even} \\ b \cdot b^{n-1} & \text{if } n \text{ is odd} \end{cases}$$

   - Implement a function `fast_expt` which computes b^n for any natural number n in $\Theta(\log n)$ time.
   - How can you extend this to integer powers?
   - Does your implementation give rise to an iterative or recursive process?
   - If iterative can you write a version that gives rise to a recursive process? If recursive can you write a version that gives rise to an iterative process?