# Report

Conor Maguire

A: The naive evaluation uses $(2N-1)N^2\left(\frac{1}{2}k(k-1)\right)$ calculations to calculate all of it's powers, $(2N-1)N^2k$ ~~calcu~~ calculations multiplying the coefficient with the powers, and $N^2(k-1)$ calculations adding everything together for a total of $(2N-1)N^2\left(\frac{1}{2}k(k-1)\right) + N^2(k-1) + N^2k(2N-1)$ operations. So the function runs in $O(N^3k^2)$ time.

B: Part III uses a max of $2k$ operations for the initial for-loop, $2k^2$ operations for the for-loop inside the main loop, $k$ for the size comparisons and $\log_2 k + \log_2 k-1 + \dots + \log_2 1 = \log_2 k!$ ~~operations for the fast powers of the~~ $*((2N-1)N^2 + 2) + k$ operations for the fast powers. $kN^2(2N-1) + kN^2$ for the multiplication and addition of coefficients to the total, and $k(2N-1)N^2$ for the final multiplications. In total we have:

$$2k + 2k^2 + \log_2 k!\,((2N-1)N^2 + 2) + k + kN^2(2N-1) + kN^2 + kN^2(2N-1)$$

So the function runs in $O(N^3k)$ time if $k > \log_2 k! \iff k! < 2^k$ or $O((\log_2 k!)N^3)$ time otherwise (if $k \geq 4$ then $k! > 2^k$)

C: If we use fast power to calculate powers in part A we ~~get~~ have $N^2(k-1) + N^2k(2N-1) + \log_2 k!((2N-1)N^2 + 2) + k$ operations. So the function runs in $O(N^3 \log_2 k!)$ time or $O(N^3k)$ time. The latter if $k > \log_2 \underline{k!}$ or the former otherwise ($k \geq 4$)

For large values of $k$ and $N$ B and C are in $O(N^3 \log_2 k!)$ time which is faster than $O(N^3k^2)$ time in part A since $|\log_2 k!| = O(k^2)$, $|\log_2 k!| \leq B|k^2|$ for $b=0$ $B=1$

Since B and C have the same order they are similarly efficient, however C requires less overall calculations since (when we look at terms with ~~the next highest~~ the next highest order.)

$N^2(k-1) + N^2k(2N-1) < 2k + 2k^2 + k + kN^2(2N-1 + 1 + 2N-1)$ ~~$N^2(2Nk-1)$~~ $\Rightarrow 2N^3k < 4N^3k$ which is true so C is the most efficient for large $n$ and $k$