

運算子+的部分為先宣告一個多項式 C，後由兩個多項式的最低次項開始，向高次項依序走訪，並比較兩多項式目前走訪到的項的次方大小，較大的一邊暫停走訪並將較小的一邊的目前項加到多項式 C，直到兩式都走訪完畢。

運算子-的部分與運算子+雷同，差異在於處理到多項式 B 時係數加上負號後再加到多項式 C。

運算子*的部分是模擬紙筆進行直式乘法的模式，但將乘數與被乘數交換，並重新排序以符合儲存格式。

微分的部分以微分基本性質 $dA/dx = d(ax^n)/dx = (a*n)x^{(n-1)}$ 為主幹，並判斷微分前是否為常數項，若是則直接捨去。

第一種求多項式數值得方法為依次走訪各項，並將結果加上該項的運算結果，若次方運算的時間複雜度等於次方數，則此方法的時間複雜度為所有項的次方數相加，但內建之數學函數 `pow` 有經過最佳化處理，時間複雜度遠低於此。

第二種求值方法為 **Horner's method**，從最高次項開始走訪，並重複乘以 `x` 值、加上相對應項的係數，時間複雜度為最高次項的次方數。

設定值函數首先判斷與輸入值同次方的項是否存在，是則直接修改係數，否則新增一相對應的項並重新排列。

快速排序函數修改自原先的整數陣列排序，同時搬移該項次方與係數，保持兩者的對應關係。

兩求值方法的時間複雜度比較(假設次方運算未經過最佳化並以基本題與加分題為例)：

暴力計算法的時間複雜度為所有項的次方相加，在基本題的時間複雜度為 $6+5+4+2+1=18$ ，加分題則為 $1006+1000+9+3=2018$ ；**Horner's method** 的複雜度則是最高次項，基本題為 6，加分題為 1006，分別降低 67%與 50%的複雜度。由以上結果可以推得若次方運算未經優化的情形下，**Horner's method** 是有效降低運算量的方式，但減少百分比與缺項數目呈現負相關。

為達成加分題功能需求，將多項式類別內新增一個陣列儲存每項次方，不再以陣列索引值作為次方，並重新編寫各運算子及函數，且修改快速排序演算法以保證所有多項是皆是以升冪儲存。