

一般來說執行 **binary search** 的時間為 $O(\log(n))$ ，但插入一個新元素到排序好的陣列中則需要線性的時間，課本中提出了一種可以降低插入新元素的時間複雜度的想法：

Let $k = \lceil \lg(n+1) \rceil$, and let the binary representation of n be $\langle n_{k-1}, n_{k-2}, \dots, n_0 \rangle$. We have k sorted arrays A_0, A_1, \dots, A_{k-1} , where for $i=0, 1, \dots, k-1$, the length of array A_i is 2^i . **Each array is either full or empty**, depending on whether $n_i=1$ or $n_i=0$, respectively. The total number of elements held in all k arrays is therefore exactly n .

看到這裡先不用緊張，題目其實就是要我們不斷將兩個 array merge 和 binary search。大家可以先想想看為什麼這種做法 insert 的時間複雜度會比較低。

詳見 CLRS problem 17-2

這禮拜的作業，請實作兩個 function：

1. merging two sorted array (array 長度可能會不一樣)
2. Binary search 找出 array 指定數字的位子