1. 了解基本的字串匹配的演算法後，請回答下列問題：

   (a) 請列出 “$mississippi$” 與 “sip” 依上述字串匹配方式的匹配過程，及求出所需要匹配的字元對數。(需要找到所有匹配的地方，而不是找到一個之後就停止)

解答

比對過程如下：

m	i	s	s	i	s	s	i	p	p	i
s										
	s									
		s	i							
			s	i	p					
				s						
					s	i				
						s	i	p		
							s			
								s		

共需比對 15 個字元。

   (b) 請敘述一種構造方式，構造出兩個長度不超過 10^6 的字串 A, B ，使得字串 B 不在字串 A 中，且依上述字串匹配方式，所需要匹配的字元對數 $\geq 10^9$，且字串 A 包含至少 10^3 種字元。

解答

注意到，如果令

$$A = \underbrace{a\cdots a}_{(k-1)\text{個}a}\, b\cdots$$
$$B = \underbrace{a\cdots a}_{k\text{個}a}$$

則用 B 比對 A 的前 k 個字元時，比對次數為 $k + (k-1) + \cdots + 1 = \frac{k(k+1)}{2}$ 次。利用此概念，令

$$A = \overbrace{\underbrace{a\cdots a}_{(k-1)\text{個}a}\, b\, \underbrace{a\cdots a}_{(k-1)\text{個}a}\, b\cdots \underbrace{a\cdots a}_{(k-1)\text{個}a}\, b}^{m\text{組}}$$
$$B = \underbrace{a\cdots a}_{k\text{個}a}$$

則 B 不在字串 A 中，且用 B 比對 A 時，一遇到 b 字元就會停下，因此可視為每組獨立比對。最後一組只會比對到第一個字元，其餘因為長度不足而不會比對到，因此我們直接忽略最後一組不計入比對次數。如此一來，可視為全部共比對了 $(m-1)$ 組，每組比對次數為 $\frac{k(k+1)}{2}$，總比對次數為 $(m-1)\times\frac{k(k+1)}{2}$。

為了滿足字串 A 包含至少 10^3 種字元，可以將其餘字元放在最後一組內，反正這些字元永遠不會用到，最後再調整 k,m 的大小滿足題目條件即可。以下提供一組合法的答案：

$$A = \overbrace{\underbrace{a\cdots a}_{9999\text{個}a}b\underbrace{a\cdots a}_{9999\text{個}a}b\cdots\underbrace{a\cdots a}_{9999\text{個}a}b}^{99\text{組}}\quad\underbrace{cdef\cdots}_{10000\text{個隨機字元}}$$
$$B = \underbrace{a\cdots a}_{10000\text{個}a}$$

比對次數為 $99\times\frac{10000\times10001}{2}\cong 4.95\times10^9\geq10^9$。

(c) 請敘述一種構造方式，構造出兩個長度不超過 10^6 的字串 A,B ，使得字串 B 不在字串 A 中，且依上述字串匹配方式，所需要匹配的字元對數 $\geq 10^9$，且字串 A 包含至少 10^3 種字元，也不存在連續相同的字元。

解答

同上一題的概念，如果令

$$A = \underbrace{ab\cdots ab}_{\frac{k-2}{2}\text{組}ab}ac\cdots$$
$$B = \underbrace{ab\cdots ab}_{\frac{k}{2}\text{組}ab}$$

則用 B 比對 A 的前 k 個字元時，比對次數為 $k+1+(k-2)+1+\cdots+2+1=\frac{k(k+4)}{4}$ 次。之後推導和上題完全相同，以下提供一組合法的答案：

$$A = \overbrace{\underbrace{ab\cdots ab}_{4999\text{組}ab}ac\underbrace{ab\cdots ab}_{4999\text{組}ab}ac\cdots\underbrace{ab\cdots ab}_{4999\text{組}ab}ac}^{99\text{組}}\quad\underbrace{defg\cdots}_{10000\text{個隨機字元}}$$
$$B = \underbrace{ab\cdots ab}_{5000\text{組}ab}$$

比對次數為 $99\times\frac{10000\times10004}{4}\cong 2.476\times10^9\geq10^9$。

2. 了解基本的排序演算法後，請回答下列問題：

(a) 請列出使用 merge sort 排序序列 $[1, 8, 5, 3, 2, 6, 4, 7]$ 的過程。

> 解答
>
> $$\begin{array}{ll} & [1,8,5,3,2,6,4,7] & \text{原始序列} \\ \Rightarrow & [\underline{1,8,5,3},\underline{2,6,4,7}] & \text{split} \\ \Rightarrow & [\underline{1,8},\underline{5,3},\underline{2,6},\underline{4,7}] & \text{split} \\ \Rightarrow & [\underline{1},\underline{8},\underline{5},\underline{3},\underline{2},\underline{6},\underline{4},\underline{7}] & \text{split} \\ \Rightarrow & [\underline{1,8},\underline{3,5},\underline{2,6},\underline{4,7}] & \text{merge} \\ \Rightarrow & [\underline{1,3,5,8},\underline{2,4,6,7}] & \text{merge} \\ \Rightarrow & [\underline{1,2,3,4,5,6,7,8}] & \text{merge} \\ \Rightarrow & [1,2,3,4,5,6,7,8] & \text{result} \end{array}$$

(b) 請列出使用 quick sort 排序序列 $[1, 8, 5, 3, 2, 6, 4, 7]$ 的過程。(pivot 可以亂選，分堆的演算法也不一定要和範例程式碼相同)

> 解答
>
> 以下參考解答為 pivot 永遠使用序列的第一個元素。
>
> $$\begin{array}{ll} & [\underline{1,8,5,3,2,6,4,7}] & \text{原始序列} \\ \Rightarrow & [1,\underline{8,5,3,2,6,4,7}] & \text{pivot=1} \\ \Rightarrow & [1,\underline{5,3,2,6,4,7},8] & \text{pivot=8} \\ \Rightarrow & [1,\underline{3,2,4},5,\underline{7,6},8] & \text{pivot=5} \\ \Rightarrow & [1,\underline{2},3,\underline{4},5,\underline{7,6},8] & \text{pivot=3} \\ \Rightarrow & [1,2,3,4,5,\underline{6},7,8] & \text{pivot=7} \\ \Rightarrow & [1,2,3,4,5,6,7,8] & \text{result} \end{array}$$

(c) 請列出使用 radix sort 排序序列 $[26, 15, 27, 35, 17, 36, 28, 16]$ 的過程。(只需要排序 2 輪即可)

> 解答
>
> $$\begin{array}{ll} & [26,15,27,35,17,36,28,16] & \text{原始序列} \\ \Rightarrow & [15,35,26,36,16,27,17,28] & \text{radix=1} \\ \Rightarrow & [15,16,17,26,27,28,35,36] & \text{radix=10} \\ \Rightarrow & [15,16,17,26,27,28,35,36] & \text{result} \end{array}$$

3. Stability 是排序演算法的一個重要性質。我們說一個排序演算法是 stable，表示對於序列中任意兩個值完全一樣的元素，在排序前後不會改變他們的相對位置，也就是不會前後互換。舉例來說，序列 $[2, 1, 2']$ 中有兩個 2，原本在後面的 2 多加了上標用以區別，如果經過排序後形成序列 $[1, 2', 2]$，則這個排序演算法就不是 stable。

   (a) 請問 merge sort, quick sort, radix sort 三個排序演算法分別是否 stable？(以範例程式碼為主)

> 解答
>
> merge sort 和 radix sort 為 stable，quick sort 不是 stable。

(b) 現在你想要排序一個資料型態為 Data 的序列，兩個該型態的物件可以用 “$<$”(小於) 運算子比較大小，若 $!(a < b)$ && $!(b < a)$，則表示 $a = b$。然而，你只能使用一個基於比較的排序函式，而這個函式使用的排序演算法並不是 stable。請想出一個方法使用這個函式，以得到一個 stable 的排序結果。

Hint: struct

> 解答
>
> 自訂一資料結構 Data2，包含原本的 Data 物件和該物件在序列中的位置。比較兩個 Data2 物件時，若兩者所包含的 Data 物件不相等，則回傳這兩個 Data 物件的大小關係，否則就回傳位置的大小關係。如此一來所有的 Data2 物件就有唯一的先後順序，不論排序演算法是否 stable，結果都會是一樣的。排序完 Data2 的序列後，再將其中的 Data 元素抽出來即可。

4. 文中的 Radix sort 是以 10 進位做為舉例，因此上文計算複雜度時，我們就將 10 當成了常數，而在某些步驟忽略了 10 造成的複雜度。但是實際上，Radix sort 並不一定要使用 10 進位，可以使用任意進位制，因此如果使用 b 進位的狀況下（簡而言之，將範例程式碼中的 10 換成 b），回答下列問題。

(a) 時間複雜度為何？

> 解答
>
> $O((n+b)\log_b C)$

(b) 額外空間複雜度為何？

> 解答
>
> $O(n+b)$

Hint: b 與 $\log b$ 都不應視為常數，並用 b, n, C 表示答案。