

Radix Sort

Ý tưởng:

- Giả sử mỗi phần tử a trong dãy a_0, \dots, a_{n-1} là số nguyên có tối đa m chữ số.
- Phân loại các phần tử lần lượt theo các chữ số hàng đơn vị, hàng chục, hàng trăm...của nó.

Các bước:

- B1: $k = 0$; //phân loại theo hàng đơn vị
- B2: Khởi tạo 10 lô B_0, \dots, B_9 rỗng (cơ chế giống stack).
- B3: Lần lượt đặt các phần tử trong mảng a vào các lô B_t với t là chữ số ở hàng k của nó.
- B4: Nối các số trong lô B theo đúng thứ tự thành a .
- B5: $k = k+1$; Nếu $k < m$ thì quay lại B2;
Ngược lại, dừng.

Radix Sort – Sắp theo cơ số

Hàng đơn vị

0	1	2	3	4	5	6	7	8	9	10	11
70 <u>1</u>	172 <u>5</u>	99 <u>9</u>	917 <u>0</u>	325 <u>2</u>	451 <u>8</u>	700 <u>9</u>	142 <u>4</u>	42 <u>8</u>	123 <u>9</u>	842 <u>5</u>	701 <u>3</u>

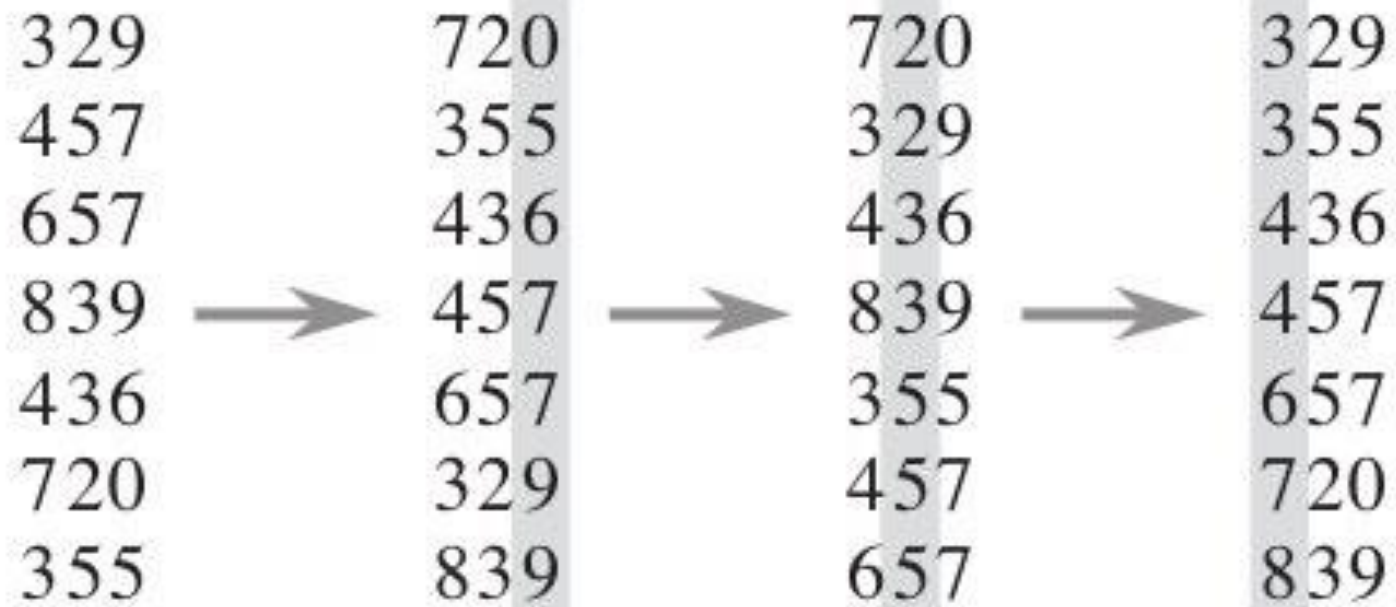
Lô B

										099 <u>9</u>	
					172 <u>5</u>				451 <u>8</u>	700 <u>9</u>	
917 <u>0</u>	070 <u>1</u>	325 <u>2</u>	701 <u>3</u>	142 <u>4</u>	842 <u>5</u>				042 <u>8</u>	123 <u>9</u>	
0	1	2	3	4	5	6	7	8	9		

0	1	2	3	4	5	6	7	8	9	10	11
9170	0701	3252	7013	1424	8425	1725	0428	4581	1239	7009	0999

Hàng ...

Radix Sort – Sắp theo cơ số



Radix Sort – Sắp theo cơ số

VD: Sắp xếp tăng dần

```
Khởi tạo B[0,...9];
for (t ← 0 to m-1) do
  for (i ← 0 to n-1) do
    Thêm a[i] vào B[Digit(a[i],t)];
  end for
  for (j ← 0 to 9) do
    Lấy ngược các phần tử từ B[j] đưa vào a;
  end for
end for
```