

DSA in-class activity TODOs

Todo 1

取得a的由右數起的第n個數位($n=1$ 個位, $n=10$ 十位, $n=100$ 百位)

Todo 2

idx: 要排序的數在value[]中的index

value: 要排序的數存放的陣列

len: 要排序的數的數量

函數完成後, idx 會按它對應的 value 的大小順序排好。

Todo 3

取得要排序的整數中最大者

Todo 4

已根據最大整數的最左數位排序過之後便會停止

Todo 5

計算有多少個數由右數起的第N位分別是0, 1, 2,..., 9

Todo 6

counter 指示排序後放回的起始位置

Todo 7

按每個數的目前數位把這個數的 idx 放在相應的 bucket 裡, 並從後往前放。

Todo 8

切換到前一個數位繼續排序

Todo 9

```
void cocktail_sort(int *idx, int *value, int len) {  
    int l = 0, r = len-1;  
    do {  
        l++;  
    } while(r);  
}
```

Bcause we need l to increment by 1 every do-while loop in order for our algorithm to work.

Todo 10

the code

```
x = x ^ y ^ (y = x)
```

is same as

```
x = x ^ y ^ x // = y  
y = x
```

this is a fancy way to swap two value using bitwise XOR.

Todo 11

we can decompose the procedure into:

```
val = x  
val ^= y  
y = x  
val ^= y  
x = val
```

```
set x = 10, y = 20, val = 0
after val = x: x = 10, y = 20, val = 10
after val ^= y: x = 10, y = 20, val = 30
after y = x: x = 10, y = 10, val = 30
after val ^= y: x = 10, y = 10, val = 20
after x = val: x = 20, y = 10, val = 20
```

Todo 12

since

```
value[idx[len-1]]
```

now holds the largest value, we dont need to check it anymore. wy only need to search from l to len-2.

Todo 13

the while loop ends when

```
r = 0
```

this is wrong because ther is no point running the while loop if

```
r <= l
```

Todo 14

```
void cocktail_sort(int *idx, int *value, int len) {
    int l = 0, r = len-1;
    do {
        //...
    } while(l<r);
}
```