1.

交换序列如下

```
70, 89, 63, 39, 87, 92, 22, 87' , 12, 6, 70' , 94, 17, 31, 11, 50
# 交换70和11
11, 89, 63, 39, 87, 92, 22, 87' , 12, 6, 70' , 94, 17, 31, 70, 50
# 交换89和31
11, 31, 63, 39, 87, 92, 22, 87' , 12, 6, 70' , 94, 17, 89, 70, 50
# 交换63和17
11, 31, 17, 39, 87, 92, 22, 87' , 12, 6, 70' , 94, 63, 89, 70, 50
# 交换87和6
11, 31, 17, 39, 6, 92, 22, 87' , 12, 87, 70' , 94, 63, 89, 70, 50
# 交换92和12
11, 31, 17, 39, 6, 12, 22, 87' , 92, 87, 70' , 94, 63, 89, 70, 50
# 最终交换pivot(50)和双指针相遇的位置(87')
11, 31, 17, 39, 6, 12, 22, 50 , 92, 87, 70' , 94, 63, 89, 70, 87'
```

2.

高位优先法 (MSD) 过程
我们从十位开始分桶，不收集，最后给出每个桶中的顺序数据。
初始序列：
$70, 89, 63, 39, 87, 92, 22, 87, 12, 6, 70, 94, 17, 31, 11, 50$

第一趟（按十位分桶）

- 十位为 0 的桶：{6}
- 十位为 1 的桶：{12, 11, 17}
- 十位为 2 的桶：{22}
- 十位为 3 的桶：{39, 31}
- 十位为 5 的桶：{50}
- 十位为 6 的桶：{63}
- 十位为 7 的桶：{70, 70}
- 十位为 8 的桶：{89, 87, 87}
- 十位为 9 的桶：{92, 94}

第二趟（对每个桶按个位分桶再排序）

每个桶按个位分桶的内容如下：

- 十位 0 的桶：无需进一步分桶，因为只含 {6}

- 十位 1 的桶：{11, 12, 17}
    - 个位 1：{11}
    - 个位 2：{12}
    - 个位 7：{17}
- 十位 2 的桶：无需进一步分桶，因为只含 {22}
- 十位 3 的桶：{39, 31}
    - 个位 1：{31}
    - 个位 9：{39}
- 十位 5 的桶：无需进一步分桶，因为只含 {50}
- 十位 6 的桶：无需进一步分桶，因为只含 {63}
- 十位 7 的桶：{70, 70}
    - 个位 0：{70, 70}
- 十位 8 的桶：{89, 87, 87}
    - 个位 7：{87, 87}
    - 个位 9：{89}
- 十位 9 的桶：{92, 94}
    - 个位 2：{92}
    - 个位 4：{94}

低位优先法 (LSD) 过程
从个位开始逐轮分桶，并在每一轮后收集数据，直到最高位。
初始序列：
$70, 89, 63, 39, 87, 92, 22, 87, 12, 6, 70, 94, 17, 31, 11, 50$

第一趟（按个位分桶）

- 个位 0 的桶：{70, 70, 50}
- 个位 1 的桶：{31, 11}
- 个位 2 的桶：{92, 22, 12}
- 个位 3 的桶：{63}
- 个位 4 的桶：{94}
- 个位 6 的桶：{6}
- 个位 7 的桶：{87, 87, 17}
- 个位 9 的桶：{89, 39}

收集结果：
{70, 70, 50, 31, 11, 92, 22, 12, 63, 94, 6, 87, 87, 17, 89, 39}

第二趟（按十位分桶）

- 十位 0 的桶：{6}
- 十位 1 的桶：{12, 11, 17}
- 十位 2 的桶：{22}
- 十位 3 的桶：{31, 39}
- 十位 5 的桶：{50}
- 十位 6 的桶：{63}
- 十位 7 的桶：{70, 70}
- 十位 8 的桶：{87, 87, 89}
- 十位 9 的桶：{92, 94}

最终结果：
$6, 11, 12, 17, 22, 31, 39, 50, 63, 70, 70, 87, 87, 89, 92, 94$

3.

采用分治算法, 假设算法为 F, 设其时间复杂度为 $T(n)$
先得到 $F(\{x_0, x_1, \ldots, x_{n-2}\})$, 将输出的结果每一项都加上 x_{n-1}, 得到数组 $F(\{x_0, x_1, \ldots, x_{n-2}\})^+$. 然后归并 $F(\{x_0, x_1, \ldots, x_{n-2}\})$ 和 $F(\{x_0, x_1, \ldots, x_{n-2}\})^+$ 即得 $F(\{x_0, x_1, \ldots, x_{n-1}\})$
归并过程在两个数组各放置一个指针进行比较, 小的一边数组值入队, 并移动指针,直到遍历完两个数组, 时间复杂度是 $O(2^{n-1})$
得到$F(\{x_0, x_1, \ldots, x_{n-2}\})$ 后其他过程的复杂度也不超过 $O(2^{n-1})$, 因此有

$$T(n) = T(n-1) + O(2^{n-1})$$

从而 $T(n) = O(2^n)$