Report

2. [5 pt] Suppose n 6-sided fair dice are rolled independently. What is the expected sum of the outcome of these dice? Please provide the derivation.

解：令 X_i 为第 i 个骰子投出的点数，我们有：

$$\mathbb{E}[\sum_i X_i] = \sum_i \mathbb{E}[X_i]$$
$$= n \times \frac{1}{6} \times (1 + \cdots + 6)$$
$$= \frac{7n}{2}$$

3.

解：

- 稳定的排序算法定义为：一个排序算法是稳定的，如果具有相等键值的两个对象在排序输出后，与它们在要排序的输入数组中以相同的顺序出现。

  插入排序、归并排序、计数排序是稳定的，快速排序是不稳定的。

- 原地算法的定义为：基本上不需要额外辅助的数据结构完成，或者说额外辅助的存储空间是与输入 n 无关的常数大小 $\mathcal{O}(c)$ 空间。

4.

解：

(a) 使用计数排序，时间复杂度为 $\mathcal{O}(n+m)$，空间复杂度为 $\mathcal{O}(n+m)$：

1. 设置从0 ~ m个桶，这里存放这相应ID的文件。可以用一次遍历找出当前文件的最小和最大的元素ID：$\mathcal{O}(n)$ 的查找复杂度。
2. 将每个文件移动到对应ID的桶内，比如文件序列是：`2，1，0，0，3，1`，那么最后的桶序列就是：`0: 2，1: 2，2: 1，3: 1`：$\mathcal{O}(n)$ 的移动复杂度。
3. 将每个桶内元素扩展成有序的排序：$\mathcal{O}(m)$ 的移动复杂度。

所以总复杂度为 $\mathcal{O}(n+m)$

(b) 要访问某一范围的文件，只需访问对应的0~m个桶映射的有序排列即可，比如(a)题中的例子，0~m个桶为：0: 2，1: 2，2: 1，3: 1

需要访问ID为 1~2 的文件时，则取出 `1: 2，2: 1` 这两个桶映射的文件，由于这里的取出桶即对应元素是 $\mathcal{O}(1)$ 的，所以时间复杂度为 $\mathcal{O}(1)$ 即可找到需要的文件们。

5.

解：

(a) 注意到所有元素是有区别的，并且 $n = 2^k, \forall k > 0$，考虑算法执行到 `Return unsuccessful`，此时比较次数达到最大，为worst case：

- 以下证明，算法第三行：`while s <= e do` 循环最多执行 $k+1$ 次：

注意到，当 `e = s` 时进入循环，有`m = s`，此时如果 `A[m] != v`，那么下一次循环即结束并跳出.

- 问题等价于证明，执行 k 次后，$e - s = 1$ 为最坏情况:

  对于 2^k 的序列，算法的while循环每执行一次，`s`和`e`之间的范围就变为原来的$\frac{1}{2}$，该性质由 `m = floor((s + e) / 2))` 可得.

  所以经过 k 次执行后，s和e之间的范围变为1，即 $k+1$ 次执行后跳出循环，最坏情况即为执行 k 次.

- 综上所述，在跳出循环之前最后一次找到该元素，则需要执行 k 次，即时间复杂度: $\mathcal{O}(k) = \mathcal{O}(\log n)$.

(b)

分析递归树，我们有:

$$
\begin{aligned}
&\textstyle\sum_{i=1}^{\log n} (\text{number of iterations in case i}) \times (\text{number of nodes in case i}) \ \div n \\
&= \textstyle\sum_{i=1}^{\log n} i \times \frac{1}{2^i} \\
&= \frac{1}{2^{\log n}} + \cdots + \frac{1}{2^2}(\log(n-1)) + \frac{1}{2}\log N \\
&\sim \mathcal{O}(logn)
\end{aligned}
$$