习题三

Hachey

3.1

数值随机算法计算数值 a 的精度可以表示为置信区间 $\Pr[x \in [a-\delta, a+\delta]] > 1-\gamma$。试利用切尔诺夫界为第 2 章计算 π 的数值随机算法之一建立置信区间，使得我们可以根据置信水平和置信区间估计所需随机实验的次数。

解：

一种计算 π 值的数值随机算法如下：

Algorithm 1: Calcπ()

Input: 一个正整数 n

Output: π 的估计值 $\hat{\pi}$

$c \leftarrow 0$;
for $i = 1$ **to** n **do**
    从 $[-1, 1]$ 中均匀随机抽取 x 和 y;
    if $x^2 + y^2 \leq 1$ **then**
        $c \leftarrow c + 1$;
    end
end
$\hat{\pi} \leftarrow 4c/n$;
return $\hat{\pi}$;

设第 i 次实验点落在圆内时，$X_i = 1$，否则 $X_i = 0$，$X = \sum_{i=1}^{n} X_i$，$\mu = \mathrm{E}[X]$，则 $\hat{\pi} = 4X/n$，$\mu = \mathrm{E}[X] = \mathrm{E}[n\hat{\pi}/4]$，由大数定律，$\mu = n\pi/4$。

下面用切尔诺夫界估计置信区间。切尔诺夫界提供了随机变量偏离其期望值的概率上界。则对于任意 $0 < \delta' < 1$，切尔诺夫界告诉我们

$$\Pr\left[|X - \mu| \geq \delta' \mu\right] < 2\mathrm{e}^{-\mu \delta'^2/3}$$

即

$$\Pr\left[|\hat{\pi} - \pi| \geq \delta' \pi\right] < 2\mathrm{e}^{-n\pi \delta'^2/12}$$

要使 $\Pr[\hat{\pi} \in [\pi-\delta, \pi+\delta]] > 1-\gamma$，即使

$$\Pr[|\hat{\pi}-\pi| > \delta] \leq \gamma$$

令 $\delta' = \delta/\pi$，则应使

$$2\mathrm{e}^{-n\delta^2/12\pi} \leq \gamma$$

解得

$$n \geq \frac{12\pi}{\delta^2} \ln \frac{2}{\gamma}$$

如此，我们可以根据置信水平和置信区间估计所需随机实验的次数。

3.2

QuickSort 排序过程可以视为算法的递归调用过程，因此整个算法的执行过程可以视为一棵递归调用树，算法的每次调用对应树中的一个结点，结点间的边表示直接嵌套的调用关系。在每次调用 QuickSort 时，首先从当前数据子集（记其大小为 s）中随机选择划分元素将当前子集划分为两个子集合；如果划分得到的两个子集的大小均不超过 $2s/3$，则称递归调用树中相应节点为*好结点*，否则称之为*坏结点*。

(a) 证明：在任意从树根到叶子的路径上，好结点的数量不超过 $c_1 \log_2 n$，其中 c_1 是一个常数；

(b) 证明：任意从树根到叶子的路径上所含结点的数量不超过 $c_2 \log_2 n$ 的概率至少为 $1-1/n^2$，其中 c_2 是一个常数；

(c) 证明：从树根到叶子的最长路径上所含结点的数量不超过 $c_2 \log_2 n$ 的概率至少为 $1-1/n$，其中 c_2 是 (b) 中的常数；

(d) 利用 (a),(b),(c) 的结论，证明：QuickSort 在 $O(n \log n)$ 时间内排序 n 个数据对象的概率至少为 $1-1/n$。

证明：

(a) 在 QuickSort 的递归调用树中，每个好结点都将使数据集的大小至少减少为原来的 $2/3$。显然，使得好结点数量最多的从树根到叶子的路径一定满足：从树根开始的若干个结点都是好结点，且这些好结点使得数据集的减少最小。设树的根结点对应的数据集大小为 n，前 k 结点为好结点，且第 $k+1$ 个结点无法成为好结点，则有

$$1 \leq n(2/3)^k < 3$$

解得

$$\frac{\log_2 \frac{n}{3}}{\log_2 \frac{3}{2}} < k \leq \frac{\log_2 n}{\log_2 \frac{3}{2}}$$

从而有好结点的数量 k 不超过 $c_1 \log_2 n$，其中 c_1 为常数 $\dfrac{1}{\log_2 \frac{3}{2}}$。

(b) 由于划分元素的选取是随机的，所以可以将每次划分是否得到好结点视为一次伯努利实验。设 p 为每次划分得到好结点的概率，则 $1-p$ 是得到坏结点的概率。由好结点的定义，我们知道 $p > 1/3$。

设 X 为从树根到叶子的路径上所含结点的数量，下证 X 不超过 $c_2 \log_2 n$ 的概率至少为 $1 - 1/n^2$。由 Chernoff 界，对于任意 $0 < \delta < 1$，有

$$\Pr\left[X \geq (1+\delta)\mu\right] < \mathrm{e}^{-\mu\delta^2/3}$$

从而有

$$\Pr\left[X < (1+\delta)\mu\right] \geq 1 - \mathrm{e}^{-\mu\delta^2/3}$$

其中 $\mu = \mathrm{E}[X]$。取 $\delta = \dfrac{c_2 \log_2 n}{\mu} - 1$，则有

$$\Pr\left[X < c_2 \log_2 n\right] \geq 1 - \exp\left(-\mu \left(\frac{c_2 \log_2 n}{\mu} - 1\right)^2 / 3\right)$$

只需证 $1 - \exp\left(-\mu \left(\dfrac{c_2 \log_2 n}{\mu} - 1\right)^2 / 3\right) \geq 1 - 1/n^2$，即证

$$\mu \left(\frac{c_2 \log_2 n}{\mu} - 1\right)^2 \geq 6 \ln n$$

易知 $\mu \geq \log_2 n$，函数 $f(x) = x\left(\dfrac{a}{x} - 1\right)^2$ 在 $x \geq a$ 时单调递增，所以只需取足够大的 c_2，使得 $(c_2 - 1)^2 \geq 6/\log_2 e$ 即可。解得 $c_2 > 3.04$。所以，任意从树根到叶子的路径上所含结点的数量不超过 $c_2 \log_2 n$ 的概率至少为 $1 - 1/n^2$。

(c) 由 (b) 知，任意从树根到叶子的路径上所含结点的数量不超过 $c_2 \log_2 n$ 的概率至少为 $1 - 1/n^2$。由于有 n 条从树根到叶子的路径，所以有至少 1 条路径上的结点数量超过 $c_2 \log_2 n$ 的概率至多为 $n \cdot 1/n^2 = 1/n$。因此，从树根到叶子的最长路径上所含结点的数量不超过 $c_2 \log_2 n$ 的概率至少为 $1 - 1/n$。

(d) 由 (c) 知，树的高度不超过 $c_2 \log_2 n$ 的概率至少为 $1 - 1/n$。在树的每一层，需要进行时间复杂度为 $O(n)$ 的操作，因此，QuickSort 在 $O(n \log n)$ 时间内排序 n 个数据对象的概率至少为 $1 - 1/n$。

3.3

设 $X_0 = 0$，而 $X_{j+1} (j \geq 0)$ 是从 $[X_j, 1]$ 均匀随机抽取的值，令 $Y_k = 2^k(1 - X_k)$。证明：序列 $Y_0, Y_1, \cdots$ 是一个鞅。

证明：

只需证明：对于任意 $k \geq 0$，有 $\mathrm{E}[Y_{k+1}|Y_0, Y_1, \cdots, Y_k] = Y_k$。由于 $Y_{k+1} = 2^{k+1}(1 - X_{k+1})$，$X_{k+1}$ 是从 $[X_k, 1]$ 均匀随机抽取的值，$X_k = 1 - Y_k/2^k$，所以 Y_{k+1} 只与 Y_k 有关。因此，有

$$
\begin{aligned}
\mathrm{E}[Y_{k+1}|Y_0, Y_1, \cdots, Y_k] &= \mathrm{E}[Y_{k+1}|Y_k] \\
&= \mathrm{E}[2^{k+1}(1 - X_{k+1})|X_k] \\
&= 2^{k+1}\left(1 - \mathrm{E}[X_{k+1}|X_k]\right) \\
&= 2^{k+1}\left(1 - \frac{X_k + 1}{2}\right) \\
&= 2^k\left(1 - X_k\right) \\
&= Y_k
\end{aligned}
$$

3.4

利用本章所学内容，分析如下随机排序算法的时间复杂性。

输入： n 个不同的值 $x_1, x_2, \cdots, x_n$；

输出： $x_1, x_2, \cdots, x_n$ 排序后的结果

步骤： 1. 从 $x_1, x_2, \cdots, x_n$ 均匀随机抽取 y_1

2. For $k = 2$ To n

3. 从 $\{x_1, \cdots, x_n\} \backslash \{y_1, y_2, \cdots, y_{k-1}\}$ 中均匀随机抽取 y_k;

4. If $y_k < y_{k-1}$ Then goto 1;

5. 输出 $y_1, y_2, \cdots, y_n$。

解：

算法的时间复杂度可以用输出时曾向 Y 中插入元素的个数来衡量。在最好的情况下，算法仅向 Y 中插入过 n 个元素，此时算法的时间复杂度为 $O(n)$。在平均情况下，由于每次都相当于是随机地打乱 X，然后检查是否有序，易知有序的概率即每次迭代成功的概率为 $1/n!$。设迭代的次数为随机变量 I，则

$$
\Pr[I = i] = \left(\frac{n! - 1}{n!}\right)^i \cdot \frac{1}{n!}
$$

I 是一个成功概率为 $1/n!$ 的几何分布，其期望值为 $\mathrm{E}[I] = n!$。所以算法的时间复杂度为 $O(n!)$。