Solutions to Quiz 2 (4th March 2025)

Question 1

Consider the following randomized algorithm. It uses `rand`(k) which returns an random integer in the range $[1, k]$ with uniform probability for each element of this finite set.

define `stopper`(a):
  Set $b = 10$.
  while $b > 0$
    Decrement b *by* 1.
    Set x *as the output of* `rand`(2).
    if $x > 1$
      return b
  return b

We will count the running time based *only* on the number of calls to `rand`.

1. In the *worst case*, what is the running time of this algorithm.
2. What is the *expected* running time of this algorithm.
3. What is the *average case* running time of this algorithm.

Solutions

Let R denote the random variable that counts the number of calls to `rand`.

The algorithm exits in one of two cases:

- x becomes bigger than 1.
- b decreases to 0.

Note that x can only be 1 or 2.

Note that the algorithm does not make use of the value of a, so we can work as if there is no input!

Part (1). In the worst case `rand`(2) returns 1 at least 10 times so that the exit happens because b becomes 0. Each time `rand` is called b is decreased by 1. Since b starts with the value 10, there are 10 calls to `rand`.

Part (2). For $i = 1, \ldots, 9$ the probability that $R = i$ is the probability that the first $i - 1$ calls to `rand` return 1 *and* the i-th call returns 2 is $(1/2)^i$. So $P[R = i] = (1/2)^i$ for $i = 1, \ldots, 9$.

By the above analysis $R \leq 10$. So the remaining case is $R = 10$. This can occur in *two* ways.

- The first 10 calls to `rand` return 1.
- The first 9 calls to `rand` return 1 and the 10-th call returns 2.

Thus, the $P[R = 10] = (1/2)^9$ (since both outcomes of the 10-th call give $R = 10$).

Thus, the expected number of calls to `rand` are

$$E(R) = \sum_{i=1}^{10} i \cdot P[R = i] = \sum_{i=1}^{9} i \cdot \frac{1}{2^i} + 10 \cdot \frac{1}{2^9}$$

Using $(1/2)^i + (1/2)^i = (1/2)^{i-1}$ and "adding from the right side":

$$\begin{aligned}
E[R] = 10 \cdot \frac{1}{2^9} + \sum_{i=1}^{9} i \cdot \frac{1}{2^i} &= \frac{1}{2^9} + \left(9 \cdot \frac{1}{2^9} + \sum_{i=1}^{9} i \cdot \frac{1}{2^i}\right) \\
= \frac{1}{2^9} + \left(9 \cdot \frac{1}{2^9} + 9 \cdot \frac{1}{2^9} + \sum_{i=1}^{8} i \cdot \frac{1}{2^i}\right) &= \frac{1}{2^9} + \left(9 \cdot \frac{1}{2^8} + \sum_{i=1}^{8} i \cdot \frac{1}{2^i}\right) \\
&= \frac{1}{2^9} + \frac{1}{2^8} + \left(8 \cdot \frac{1}{2^8} + \sum_{i=1}^{8} i \cdot \frac{1}{2^i}\right) \\
&= \frac{1}{2^9} + \frac{1}{2^8} + \cdots + \frac{1}{2^k} + \left(k \cdot \frac{1}{2^k} + \sum_{i=1}^{k} i \cdot \frac{1}{2^i}\right) \\
&= \frac{1}{2^9} + \frac{1}{2^8} + \cdots + \frac{1}{2} + \left(\frac{1}{2} + \frac{1}{2}\right) \\
&= 2 - \frac{1}{2^9} = \frac{1023}{512}
\end{aligned}$$

Part (3). We run the algorithm a large number N of times to calculate the average number of calls to `rand` as

$$C_N = \sum_{i=1}^{10} \frac{i \cdot N_i}{N}$$

where N_i out of N is the number of times when there are i calls to `rand`.

By the law of large numbers we know that C_N *almost surely* converges to $E[R]$ as N goes to infinity.

Thus, $E[R]$ can *perhaps* be seen as the average case running time.

However, this argument is a bit specious! We usually talk about average case complexity *only* for deterministic (non-random) algorithms. For such algorithms, we *average* the running time over all possible inputs to calculate the average case complexity.

In other words, suppose the possible inputs for an algorithm A lie in a finite set S. We calculate the running time R_s for each input $s \in S$. (Since A is deterministic, R_s is a *definite* number.) We then calculate the average $\left(\sum_{s \in S} R_s\right)/|S|$ to get the average case running time of A.