1. Let 0 < α < .5 be some constant (independent of the input array length n). Recall the Partition subroutine employed by the QuickSort algorithm, as explained in lecture. What is the probability that, with a randomly chosen pivot element, the Partition subroutine produces a split in which the size of the smaller of the two subarrays is ≥ α times the size of the original array?

The algorithm picks a pivot at random, therefore each element has an equal probability of getting picked, i.e. $\frac{1}{n}$.

Let's create an example scenario to make the question easier to picture. If n = 40 and α= 0.25, then there are two scenarios:

1. The pivot is one of the smallest n * α elements in the array, then it'd create a partition that is less than α times the size of the original array. So it must be larger than or equal to the 10th element (n * α = 40 * 0.25 >= 10).
2. The pivot is one of the largest n * α elements in the array, then it'd also create a partition that is less than α times the size of the original array. So it must be less than or equal to the 30th element.

Therefore, there are n - 2 * n*αpossible elements to choose from that will fit the criteria, each with $\frac{1}{n}$ probability.

$$\frac{1}{n}(n - 2 * n * \alpha) = 1 - 2\alpha$$

1 - 2α

2. Now assume that you achieve the approximately balanced splits above in every recursive call --- that is, assume that whenever a recursive call is given an array of length k, then each of its two recursive calls is passed a subarray with length between αk and (1−α)k (where α is a fixed constant strictly between 0 and .5). How many recursive calls can occur before you hit the base case? Equivalently, which levels of the recursion tree can contain leaves? Express your answer as a range of possible numbers d, from the minimum to the maximum number of recursive calls that might be needed.

Using the previous example again with k = 40 and α= 0.25, then each of the recursive calls is passed a subarray with length between α*k = 10 and (1 - α)*k = 30.

If α< ½ , then 1 - α > α (0.75 > 0.25). Therefore, the subarray with the greatest recursion depth is n*(1 - α).

Let's say d is the height of the recursion tree. Then,

$$n(1 - \alpha)^d = 1$$

Take $log_{1-\alpha}$ on both sides

$$log_{1-\alpha}(n * (1 - \alpha)^d) = log_{1-\alpha}(1)$$

$$log_{1-\alpha}(n) + d * log_{1-\alpha}(1 - \alpha) = log_{1-\alpha}(1)$$

$$log_{1-\alpha}(n) + d = 1$$

$$log_{1-\alpha}(n) = - d \quad *log_c(x) = log_b(x) / log_b(c)$$

$$d = -\frac{logn}{log(1-\alpha)}$$

The smallest d occurs when we recurse on the smaller subarray. So,

$n(\alpha^d) = 1$

$log(\alpha^d n)$= 0 $\quad$ $*log_a(xy) = log_a(x) + log_b(y), log_b(M^p) = plog_b(M)$

$d\, log(\alpha) + log(n)$= 0

$d = -\frac{log(n)}{log(\alpha)}$

Worst case: $-\frac{logn}{log(1-\alpha)}$

Best case: $-\frac{log(n)}{log(\alpha)}$

3. Define the recursion depth of QuickSort to be the maximum number of successive recursive calls before it hits the base case --- equivalently, the number of the last level of the corresponding recursion tree. Note that the recursion depth is a random variable, which depends on which pivots get chosen. What is the minimum-possible and maximum-possible recursion depth of QuickSort, respectively?

The minimum occurs when we pick the median all the time and the maximum possible depth occurs when we pick the smallest element all the time.

Minimum: $\theta(logn)$, Maximum: $\theta(n)$

4. Consider a group of k people. Assume that each person's birthday is drawn uniformly at random from the 365 possibilities. (And ignore leap years.) What is the smallest value of k such that the expected number of pairs of distinct people with the same birthday is at least one? [Hint: define an indicator random variable for each ordered pair of people. Use linearity of expectation.]

Total number of people = k

With d = 365, the probability that a particular pair share a birthday is $\frac{1}{d}$.

We have k people, so there are (k choose 2) = $\frac{k*(k-1)}{2}$ pairs of people.

The expected number of pairs will be $\frac{k*(k-1)}{2} * \frac{1}{d}$

According to the question, we want the probability where $\frac{k*(k-1)}{2} * \frac{1}{365} \geq 1$

$k^2 - k - 730 \geq 0 \; -> \; (k - 28) * (k - 27)$ = 0 (approximation)

k = 27.02 so 28

k = 28

5. Let X_1, X_2, X_3 denote the outcomes of three rolls of a six-sided die. (I.e., each X_i is uniformly distributed among 1,2,3,4,5,6, and by assumption they are independent.) Let Y denote the product of X_1 and X_2 and Z the product of X_2 and X_3. Which of the following statements is correct?

A and B are independent iff P(A * B) = P(A) * P(B)

Therefore, by definition, the following two options cannot be correct:

- Y and Z are independent, but E[Y∗Z] ≠ E[Y] ∗ E[Z]
- Y and Z are not independent, but E[Y∗Z] = E[Y] ∗ E[Z]

For the following, "Y and Z are independent, and E[Y∗Z] = E[Y] ∗ E[Z]", Y and Z are NOT independent. Because they both rely on X_2

Y and Z are not independent, and E[Y∗Z] ≠ E[Y] ∗ E[Z]