Class 02/01, Probabilities, Expected Values, IRV: Solutions

---

1. Probabilities

a. Let *X* be a random variable that takes values {5, 10, 15} with the following probabilities Pr[*X*=5] = $\frac{3}{5}$, Pr[*X*=10] = $\frac{1}{5}$, Pr[*X*=15] = $\frac{1}{5}$. Compute E[X], the expected value of *X*. Show your work.

Answer: The definition of the expectations states:

$$\begin{aligned} E[X] &= \sum_x x \cdot Pr[X = x] = \sum_{x=\{5,10,15\}} x \cdot Pr[X = x] \\ &= 5 \cdot Pr[X = 5] + 10 \cdot Pr[X = 10] + 15 \cdot Pr[X = 15] \\ &= 5 \cdot 3/5 + 10 \cdot 1/5 + 15 \cdot 1/5 = 8 \end{aligned}$$

b. Let Y be a random variable that takes values {1, 2, 3, 4, 5, 6} with equal probabilities. Compute E[*Y*], the expected value of *Y*. Show your work.

Answer: Since Y takes the six values with equal probabilities, the probability that Y takes any one of those values is $\frac{1}{6}$. Then by definition of expectations:

$$E[Y] = \sum_{y=1}^{6} y \cdot Pr[Y = y] = \sum_{y=1}^{6} y \cdot 1/6 = 1/6 \cdot \frac{6 \cdot 7}{2} = 3.5$$

This is the average of all numbers that a fair dice will show after many rolls.

2. Permutation search

Let *A* be an array of size *n* that contains integers 1 through *n*, which are ***randomly*** permuted.

Here is an algorithm that takes as input the array *A* and an integer *k*, where $1 \le k \le n$, and returns the index *i* such that A[i] = k.

```
int search (A, k)
    for i = 1 to n
       if A[i] == k
          return i
```

Analyze the expected running time of your algorithm as follows.

a. Define a random variable X_i = I{ *A*[*i*] = *k* }. What is E[X_i] ?

Answer: E[X_i] = Pr[X_i = 1], i.e. Pr[*A*[*i*] == *k*]. Since elements of *A* are randomly permuted, *k* is equally likely to be in either of the *n* positions of *A*. Thus, the E[X_i] = Pr[*A*[i] == k] = 1/n. Another way to see this, is the following: There are n! possible permutations of n elements. If we fix A[i] = k, then there are (n-1)! way to permute the remaining (n-1) elements. Thus, (n-1)!/n! = 1/n fraction of all possible n! permutations, where A[i] = k.

b. Let *Y* be a random variable that denotes the number of elements checked by linear search when searching for key *k*. Determine the expression for *Y* in terms of X_i's.

Answer: If A[i] = k, then the above algorithm will return after i steps. Thus, $Y = \sum_{i=1}^{n} i \cdot X_i$.

c. Use parts (i) and (ii) to compute the expected runtime of your algorithm. *(Remember that E[a*X] = a*E[X] for any constant a and random variable X)*

Answer: $E[Y] = E\left[\sum_{i=1}^{n} i \cdot X_i\right]$.

By linearity of expectations:

$$= \sum_{i=1}^{n} E\left[i \cdot X_i\right] = \sum_{i=1}^{n} i \cdot E[X_i] = \sum_{i=1}^{n} i \cdot \frac{1}{n} = \frac{1}{n} \cdot \sum_{i=1}^{n} i = \frac{1}{n} \cdot \frac{n(n+1)}{2} = \frac{n+1}{2}$$

Thus, the above algorithm will traverse about half of the elements on average.

Extra Credit. *Finished early? Work on the following problem*

3. Let Y be a random variable that takes values {1, 2, 3, … n} with equal probabilities. Compute E[*Y*], the expected value of *Y*. Show your work. (This is the expected value of an n-sided dice.)

Following 1b,

$$E[Y] = \sum_{y=1}^{n} y \cdot Pr[Y = y] = \sum_{y=1}^{n} y \cdot 1/n = \frac{n(n+1)}{2n}$$

If you have m n-sided die, what would be the expected value of rolling all m of them?

Multiply the above by m.