

CS3343 Analysis of Algorithms Fall 2017

Homework 3

Due 9/24/17 before 11:59pm (Central Time)

1. Probability (7 points)

For the following problem, clearly describe the sample space and the random variables you use. Be sure to justify where you get your expected values from.

Consider playing a game where you roll n fair six-sided dice. For every 1 or 6 you roll you win \$30, for rolling any other number you lose \$3n.

- (1) First assume $n = 1$ (i.e., you only roll one six-sided die).
- (a) (1 point) Describe the sample space for this experiment.

$$\boxed{S = \{1, 2, 3, 4, 5, 6\}} \rightarrow \text{One dice with 6 possible outcomes.}$$

- (b) (1 point) Describe a random variable which maps an outcome of this experiment to the winnings you receive.

Let X be a random variable on the sample space S . If $s \in S$, then $X(s)$ is a real number.

$$X(1) = X(6) = \$30$$

$$X(2) = X(3) = X(4) = X(5) = -3n = -\$3$$

- (c) (1 point) Compute the expected value of this random variable.

$$\begin{aligned} E(X) &= \sum_{s \in S} p(s)X(s) \\ &= p(1)X(1) + p(2)X(2) + p(3)X(3) + p(4)X(4) + p(5)X(5) + p(6)X(6) \\ &= (1/6)(30) + (1/6)(-3) + (1/6)(-3) + (1/6)(-3) + (1/6)(-3) + (1/6)(30) \\ &= (1/6)(30 - 3 - 3 - 3 - 3 + 30) \\ &= (1/6)(48) \\ &= 8 \end{aligned}$$

$$\boxed{E(X) = \$8}$$

- (2) Now assume $n = 6$ (i.e., you roll six six-sided dice).
 (a) (1 point) Describe the sample space for this experiment (you don't need to list the elements but describe what is contained in it).

The sample space will contain all possible combinations of six digits, each with 6 possible outcomes. $6^6 = 46656$ possible combinations.
 $S = \{111111, 111112, 111113, 111114, \dots, 666664, 666665, 666666\}$

- (b) (1 point) Describe a random variable which maps an outcome of this experiment to the winnings you receive. (Hint: Express your random variable as the sum of six random variables.)

Let X be a random variable on the sample space S . If $s \in S$, then $X(s)$ is a real number.

$$X = X_1 + X_2 + X_3 + X_4 + X_5 + X_6$$

$$= \sum_{i=1}^n X_i$$

- (c) (1 point) Compute the expected value of this random variable using the linearity of expectation. Based on this would you play this game?

$$\begin{aligned} E(X) &= E(X_1) + E(X_2) + \dots + E(X_6) \\ &= \sum_{i=1}^n E(X_i) \end{aligned}$$

$$\begin{aligned} E(X_1) &= p(1)X_1(1) + p(2)X_1(2) + p(3)X_1(3) + p(4)X_1(4) + p(5)X_1(5) + p(6)X_1(6) \\ &= (1/6)(30) + (1/6)(-18) + (1/6)(-18) + (1/6)(-18) + (1/6)(-18) + (1/6)(30) \\ &= (1/6)(30 - 18 - 18 - 18 - 18 + 30) \\ &= -\$12 \end{aligned}$$

$$E(X_1) = E(X_2) = E(X_3) = E(X_4) = E(X_5) = E(X_6)$$

$$E(X) = (-12) + (-12) + (-12) + (-12) + (-12) + (-12)$$

$E(X) = -\$72$ → If this is the correct possible outcome, I would definitely not play the game.

- (3) (1 point) What is the largest value of n for which you would still want to play the game? Justify your answer.

The largest value of n to break even would be $n = 5$

$$E(X) = (1/6)(30) + (1/6)(-3n) + (1/6)(-3n) + (1/6)(-3n) + (1/6)(-3n) + (1/6)(30)$$

$$E(X) = (1/6)(30) + (1/6)(-3(5)) + (1/6)(-3(5)) + (1/6)(-3(5)) + (1/6)(-3(5)) + (1/6)(30)$$

$$E(X) = (1/6)(30) + (1/6)(-15) + (1/6)(-15) + (1/6)(-15) + (1/6)(-15) + (1/6)(30)$$

$$E(X) = (1/6)(30 - 15 - 15 - 15 - 15 + 30)$$

$$E(X) = \$0$$

To play the game, I would play with any value of $n = 4$ or lower because with $n = 5$ I wouldn't make any money. With any value of $n = 6$ or more, the amount of loss from rolling a 2 - 6 is greater than the possible winning amount from rolling a 1 or 6.

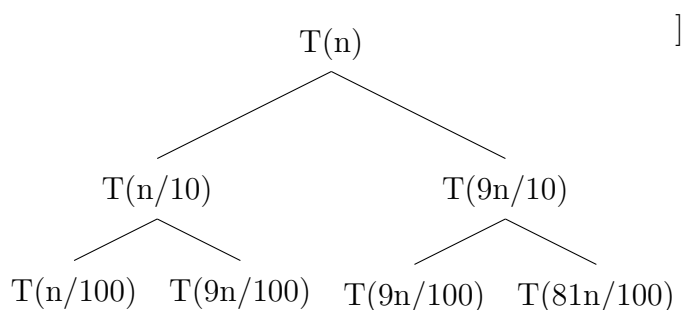
2. Variants of quicksort (6 points)

Suppose company X has created 3 new variants of quicksort but, because they are unsure which is asymptotically best, they have hired you to analyze their runtimes. Compute the asymptotic runtimes of the variants and use this to justify which is best:

- Variant 1: Partitioning the array still takes $\Theta(n)$ time but the array will always be divided into (at least) a 10% and 90% portion.

Solution :

$$T(n) = T(n/10) + T(9n/10) + \Theta(n)$$



Tree continues until $T(1)$ is reached. With how the partitioning is occurring, the left side of the tree will reach the base case before the right side.

The height of the left side of the tree: $height = (n/10^h) = 1$

$$n = 10^h$$

$$h = \log n$$

For the right side: $height = \frac{n}{(\frac{10}{9})^h} = 1$

$$n = (10/9)^h$$

$$h = \log_{(10/9)} n$$

$$\Rightarrow \boxed{T(n) \in \Theta(n \log n)}$$

- Variant 2: Partitioning the array now takes $\Theta(n^{1.1})$ time but the array will always be divided perfectly in half.

Solution:

$$T(n) = 2T(n/2) + \Theta(n^{1.1})$$

Master Theorem:

$$a = 2, b = 2, f(n) = n^{1.1}$$

$$n^{\log_b a} = n^{\log_2 2} = n$$

Case 3:

$$n^{1.1} \in \Omega(n^{1+\varepsilon}), \varepsilon = 0.1$$

$$n^{1.1} \in \Omega(n^{1.1}) \text{ Trivially True}$$

Now Show: $af(n/b) \leq cf(n)$ for $c < 1$

$$2f(n/2) \leq cn^{1.1}$$

$$2(n/2)^{1.1} \leq cn^{1.1}$$

$$2(n^{1.1}/2.14) \leq cn^{1.1} \text{ when } c = (2/2.14) < 1$$

$$\Rightarrow \boxed{T(n) \in \Theta(n^{1.1})}$$

- Variant 3: Partitioning the array now only takes $\Theta(\sqrt{n})$ time but all of the numbers except the pivot will be partitioned into a single array.

$$T(n) = T(0) + T(n-1) + \Theta(\sqrt{n})$$

$$= \Theta(1) + T(n-1) + \Theta(\sqrt{n})$$

$$= T(n-1) + \Theta(\sqrt{n})$$

Hint: It may help to know that $\sum_{i=1}^n \sqrt{i} \in \Theta(n^{3/2})$ and that $\log n \in o(n^c)$ for all $c > 0$.

3. Linear search (6 points)

The code below searches an unsorted array, A , for a given value. It returns the index of the value if it is present. Otherwise it returns -1 .

Algorithm 1 `int linearSearch(int $A[0 \dots n - 1]$, int val)`

```
 $i = 0;$ 
while  $i < n$  do
    if  $val == A[i]$  then
        return  $i$ ;
    end if
     $i++$ ;
end while
return  $-1$ ;
```

Suppose before we call `linearSearch` we randomly permute (i.e., reorder) the elements in A . In the following problem, we will find the expected asymptotic runtime of `linearSearch`.

- (1) (1 point) What is the best case asymptotic runtime of `linearSearch` on an array of n elements? What index must val be located at to elicit this runtime?

$\Theta(1)$ \rightarrow The element must be in the first index of the array.

- (2) (1 point) What is the worst case asymptotic runtime of `linearSearch` on an array of n elements? What index must val be located at to elicit this runtime?

$\Theta(n)$ \rightarrow The element must be in the last index or not exist in the array.

- (3) (1 point) For the sake of simplicity, assume val is in A . Let X_j be the random variable which, given an permutation of the elements in our array, is 1 if index $A[j]$ will be checked in our linearSearch for that permutation and 0 otherwise. What is the expected value of X_j ?

$$X_j(s) = \begin{cases} 1 & \text{if } A[j] \text{ will be checked} \\ 0 & \text{Otherwise} \end{cases} \quad (1)$$

$$E(X) = \sum_{j=1}^n E(X_j)$$

$$E(X_j) = (1/j)(1) + (1/j)(0)$$

$$E(X_j) = (1/j)$$

$$E(X) = \sum_{j=1}^n (1/j)$$

- (4) (1 point) Let X be the random variable which, given an permutation of the elements in our array, is the number of elements of A which will be checked in our linearSearch for that permutation. Define X using n copies of the random variable defined in the previous step.
- (5) (2 points) Compute the expected value of X for using the linearity of expectation. Given this, what is the expected asymptotic runtime of linearSearch?