CSCI 4470 Algorithms

Part I Foundations

Chapter 5: Probabilistic Analysis and Randomized Algorithms

5.1 Probabilistic Analysis 5.1 概率分析

- **Definition**: Analyzing algorithms in terms of average performance over a distribution of inputs.
- **Indicator Random Variables**: Useful tool in probabilistic analysis, helping in calculating the expected value directly.

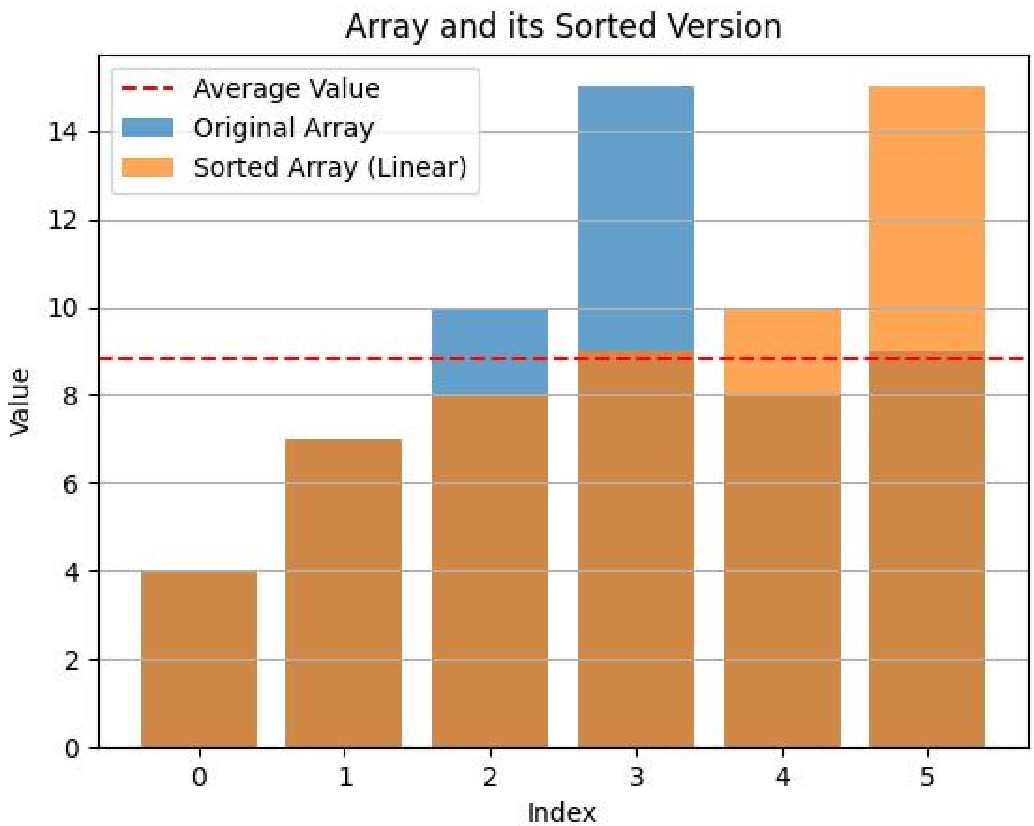
Probabilistic Analysis of Sorting Algorithm

- **Uniform Distribution**:
  - **Definition**: Every possible outcome is equally likely.
  - **In Context**: When analyzing the sorting algorithm, we assume that each permutation of the input array is equally likely.

Example: Sorting the Array

- **Initial Array**: {4,7,10,15,8,9}
- **Graph Representation**: Visual representation of the array can help in understanding the distribution and arrangement of elements.
- **Average-Case Analysis**:
  - **Step 1**: Identify all possible permutations of the array.

- **Step 2**: For each permutation, calculate the number of comparisons or operations needed to sort the array.
- **Step 3**: Find the average number of operations over all permutations to get the average-case time complexity.



1. **Average Case Analysis**
   - **Uniform Distribution**: Assuming that each input has an equal probability of occurring. In the context of your array example {4,7,10,15,8,9}, we can say it's uniformly distributed if it is equally likely to be in any permutation of its sorted order.
   - **Average Value**: It is the sum of all the values divided by the number of values. For your array, it would be $\frac{4+7+10+15+8+9}{6} = \frac{53}{6} \approx 8.83.$
2. **Randomized Algorithms**
   - **Randomized QuickSort**: An example of a randomized algorithm where the pivot element is chosen randomly.
   - **Randomized Input**: Sometimes, the algorithm randomizes the input to ensure a good average case performance.
3. **Indicator Random Variables**
   - **Usage in Analysis**: Helps in simplifying the analysis of the average case by breaking down the problem into simpler events and using indicator random variables to indicate the occurrence of these events.
   - **Example**: In the context of sorting, an indicator random variable can be used to represent the event where a particular element is in its correct position after the sorting is done.

Example Analysis with Your Array

- **Array**: {4,7,10,15,8,9}
- **Average Value**: 8.83

- **Uniform Distribution**: Assuming any permutation of the array is equally likely, we can analyze the sorting algorithm's behavior over all possible permutations to find the average case performance.

Discussion

- **Uniform Distribution Assumption**: This assumption allows us to calculate the average-case complexity by considering all possible inputs and their equal likelihood.
- **Practical Implication**: Understanding the average-case complexity helps in predicting the algorithm's performance on an average input, providing a realistic view of the algorithm's efficiency.

5.2 Randomized Algorithms

- **Definition**: Algorithms that make random choices during their execution to ensure good average performance.
- **Randomized Algorithms Types**:
  - **Las Vegas**: Always correct, runtime is probabilistic.
  - **Monte Carlo**: Runtime is deterministic, correctness is probabilistic.

Random Variable Notation

Sample Space 樣本空間 S:

- The set of possible events

Indicator Random Variable for any given event $A \in S$:

- $I(A) = 1$ if A occurs
- $I(A) = 0$ if A does not occur
- Also denoted X_A

$Pr\{A\}$: Probability that A will occur

Example: A coin toss

- $S = \{HH, HT, TH, TT\}$, where S is the random variable, and HH is the event $\{A\}$
- $Pr\{HH\} = \frac{1}{4} = \frac{1}{2} \times \frac{1}{2}$

Axioms of Probability

A **Probability Distribution** $Pr\{\}$ on a sample space S is a mapping from S to $\mathbb{R}$ satisfying:

- $Pr\{A\} \geq 0$ for any event $A \in S$
- $Pr\{S\} = 1$
- $Pr\{A \cup B\} = Pr\{A\} + Pr\{B\}$ for any two mutually exclusive events A and B

Probability Identities

- $Pr\{\emptyset\} = 0$
- If $A \subseteq B$ then $Pr\{A\} \leq Pr\{B\}$
- $Pr\{S - A\} = 1 - Pr\{A\}$
- For any **Two Events** A and B, $Pr\{A \cup B\} = Pr\{A\} + Pr\{B\} - Pr\{A \cap B\}$
- **Conditional Probability**:
  - If $Pr\{B\} > 0$, then $Pr\{A|B\} = \frac{Pr\{A \cap B\}}{Pr\{B\}}$
  - We can say it's the probability of event A given that event B has occurred
- Events A and B are independent if and only if $Pr\{A \cap B\} = Pr\{A\} \cdot Pr\{B\}$, equivalently $Pr\{A|B\} = Pr\{A\}$.

Example: The probability of getting a six given that a six occurs on the first throw of a die?

Step 1: Define the Events

- Event A: Getting a six on a throw
- Event B: Getting a six on the first throw

Step 2: Find $P(A)$ and $P(B)$

- $P(A)$: The probability of getting a six in a single throw of a die is $\frac{1}{6}$
- $P(B)$: The probability of getting a six on the first throw is also $\frac{1}{6}$

Step 3: Find $P(A \cap B)$

- $P(A \cap B)$: Since event B is a subset of event A, $P(A \cap B) = P(B) = \frac{1}{6}$

Step 4: Apply the formula

- Probability of event A given that event B has occurred
- Using the formula $P(A|B) = \frac{P(A \cap B)}{P(B)}$, we substitute the values we found in steps 2 and 3:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{\frac{1}{6}}{\frac{1}{6}} = 1$$

Example The probability of rolling two dice and getting two sixes

Step 1 步驟 1: Define the events

- Event A: Getting a six on the second die
- Event B: Getting a six on the first die

Step 2 步驟 2: Find individual probabilities

- $P(B)$: Probability of getting a six on the first die
  - $P(6_1)$ (the probability of getting a six on the first die) is $\frac{1}{6}$.
- $P(A \cap B)$: Probability of both dice showing a six: $\frac{1}{6} \times \frac{1}{6} = \frac{1}{36}$
  - $P(6_2 \cap 6_1)$ (the probability of getting a six on both dice) is $\frac{1}{6} \times \frac{1}{6} = \frac{1}{36}$.

Step 3: Apply Conditional Probability Formula

- Using $P(A|B) = \frac{P(A \cap B)}{P(B)}$
  - By the Identity $P(A \bigcap B) = P(A) \cdot P(B) = \frac{1}{6} \times \frac{1}{6} = \frac{1}{36}$

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{\frac{1}{36}}{\frac{1}{6}} = \frac{1}{6}$$

$$P(6_2|6_1) = \frac{P(6_2 \bigcap 6_1)}{P(6_1)} = \frac{\frac{1}{36}}{\frac{1}{6}} = \frac{1}{6}$$

5.3 Examples of Randomized Algorithms

Common examples and their key characteristics:

- **Hiring Problem**:
  - **Probabilistic Analysis**: Analyzing the average case of the number of hires.
  - **Randomized Algorithm**: Randomly permuting the order of applicants to get an average-case performance from the worst-case input.
- **QuickSort**:
  - **Randomized Version**: Picking a random pivot to get a good average runtime.
  - **Analysis**: Analyzing the expected number of comparisons.
- **Hashing**:
  - **Universal Hashing**: Randomly selecting a hash function to reduce the chance of collision.
  - **Perfect Hashing**: Two-level hashing scheme with no collisions at the second level.

5.4 Probabilistic Analysis and Further Uses

- **Probabilistic analysis in data structures**: Using probabilistic analysis to maintain dynamic tables optimally.
- **Probabilistic methods in graph algorithms**: Leveraging randomness to find a good approximation of a solution quickly.

Key Concepts and Terms

- **Expectation**: The average outcome of a random variable.

- **Random Variable**: A variable that can take on different values with different probabilities.
- **Event**: A subset of a sample space.
- **Probability**: A measure of the likelihood of an event occurring.

1. **Probabilistic Analysis**
   - **Definition**: Analyzing algorithms in terms of the average case rather than the worst case. It involves determining the distribution of the input space.
   - **Application in Sorting Algorithms**: Understanding how the average case performance of a sorting algorithm can be much better than the worst case performance.
2. **Randomized Algorithms**
   - **Definition**: Algorithms that make random choices during their execution to ensure good average performance.
   - **Application in Sorting Algorithms**: Randomized algorithms can be used in sorting to avoid the worst-case scenario by randomizing the input or the algorithm itself.
3. **Indicator Random Variables**
   - **Definition**: Random variables used to indicate the occurrence of an event, facilitating the calculation of the expected value.
   - **Application in Sorting Algorithms**: Can be used to analyze the average case performance of a sorting algorithm by defining suitable events and calculating their probabilities.

Expected Value

- **Definition**:
  - $E[X]$: Average outcome in a large number of experiments.
  - Note: May not be a real outcome.
- **Calculation**:
  - **Value function**, $v()$: Assigns a real number to each outcome.
  - **Formula**:

$$E[X] = \sum_{A \in S} (v(A) \cdot Pr\{A\})$$

    Where:
    - S: All possible outcomes, A: An event, $v(A)$: Value of event A, and $Pr\{A\}$: Probability of A.
- **Application**:
  - Used in Chapter 5 for probabilistic algorithm analysis.

Expected Value of a Random Variable

Definition

- **Random Variable,** X: A variable whose outcome is determined by a random experiment.

- **Expected Value, $E[X]$**: The average value of X over many trials.

Example

- **Sample Space, S**: All possible outcomes of an experiment.
  - $S = \{H, T\}$ (Heads or Tails in a coin toss)
- **Probability Distribution, $P(X)$**
  - $P(H) = \frac{1}{2}$
  - $P(T) = \frac{1}{2}$

Calculation

- **Formula**

$$E[X] = \sum_{x \in S} x \cdot P(X = x)$$

  - Where x is a value in the sample space S.

Example: A bag of gems, 5 quartz ($5), 4 amethyst ($30), 3 ruby ($80), 2 diamond ($100), 1 emerald ($130), If I grab a gem from the bag, what value should I expect?

Step 1: Identify the Random Variable

- **Random Variable, X**: Value of the gem grabbed

Step 2: Find the Probability of Each Outcome

- **Total gems**: 15
- **Probability of grabbing each type of gem:**
  - Quartz: $\frac{5}{15}$, Amethyst: $\frac{4}{15}$, Ruby: $\frac{3}{15}$, Diamond: $\frac{2}{15}$, and Emerald: $\frac{1}{15}$.

Step 3: Calculate the Expected Value

Using the formula for expected value

$$E[cost|price] = \sum_{\text{gems}} g_i P(g_i)$$

Substitute the values:

$$E(X) = \left(5 \cdot \frac{5}{15}\right) + \left(30 \cdot \frac{4}{15}\right) + \left(80 \cdot \frac{3}{15}\right) + \left(100 \cdot \frac{2}{15}\right) + \left(130 \cdot \frac{1}{15}\right)$$

Step 4: Solve

Calculate the sum:

$$E(X) = \left(\frac{25 + 120 + 240 + 200 + 130}{15}\right) = \frac{715}{15} \approx 47.67 \text{ USD}$$

Result

- **Expected value**: 47.67 USD

You should expect the value of a gem you grab to be approximately 47.67 USD on average.

Note

The formula $E[cost|price] = \sum_{\text{gems}} g_i \cdot P(g_i)$ is a fundamental concept in probability theory, representing the expected value or the average outcome over many runs of the experiment. It is derived from the individual probabilities and values of each gem.

Linearity of Expectation

Definition

- **Expectation**: The average outcome over many trials.

Properties

1. **Addition**

$$E[x + y] = E[x] + E[y]$$

   - **Note**: x and y do not need to be independent.
2. **Multiplication**

$$E[x \cdot y] = E[x] \cdot E[y]$$

   - **Note**: Valid only when x and y are independent.

Application in Chapter 5

- **Algorithm Analysis**: Utilized in probabilistic analysis to predict average outcomes.

Example Consider a problem of flipping two coins you earn $3 both heads and loose $2 for both tails and earn $1 for other case, What is the expected earning?

Step 1: Define the Random Variable

- **Random Variable, X**: The earnings from flipping two coins

Step 2: Identify the Sample Space and Probabilities

- **Sample Space,** S: $\{HH, HT, TH, TT\}$
- **Probabilities**: $P(HH) = \frac{1}{4}$, $P(HT) = \frac{1}{4}$, $P(TH) = \frac{1}{4}$, and $P(TT) = \frac{1}{4}$

Step 3: Define the Value Function

- **Value Function,** $v()$ or The cost x:
  - $v(HH) = 3$ USD, $v(HT) = 1$ USD, $v(TH) = 1$ USD, and $v(TT) = -2$ USD

Step 4: Calculate the Expected Value

Using the formula for expected value:

$$E[X] = \sum_{A \in S} (v(A) \cdot P(A))$$

Substitute the values:

$$E[X] = \left(3 \cdot \frac{1}{4}\right) + \left(1 \cdot \frac{1}{4}\right) + \left(1 \cdot \frac{1}{4}\right) + \left(-2 \cdot \frac{1}{4}\right)$$

Step 5: Solve

Calculate the sum:

$$E[X] = \left(\frac{3+1+1-2}{4}\right) = \frac{3}{4} \text{ USD} \approx 0.75 \text{ USD}$$

Result

- **Expected Earning**: 0.75 USD

You should expect to earn approximately 0.75 USD on average per game.

Example The hiring problem, Your company is hiring an assistant and you need to fill the position quickly. You hire an employment agency. They send one candidate each day and charge a small fee (c_1) for each candidate. Hiring a candidate has an additional associated cost (c_2). You want to have the best candidate at all times. After you interview each candidate, if s/he is better than your current assistant, you fire the current one and hire the interviewee.

```
Hire-Assistant(n)
1. best = 0 # Initialize with no best candidate
2. for i = 1 to n  # Loop through all candidates
3.   interview candidate i  # Interview the current candidate
4.   if candidate i is better than best # Check if current candidate is better
5.      if best > 0 fire best # If there's a current assistant, fire them
6.      best = i # Update the best candidate
7.      hire candidate i # Hire the current candidate
```

- This algorithm ensures that you always have the best candidate hired by the end of the process.

Step 1: Define the Variables

- n: Number of candidates
- c_1: Cost of interviewing a candidate
- c_2: Cost of hiring a candidate
- m: Expected number of hires, given by H_n
- Total Cost $= C_1 \cdot n + C_2 \cdot H_n(\ or\ m)$

Step 2: Define the Random Variable

Best and Worst Cases:

- **Best Case**: Hire the first candidate.
  - Cost: $c_1 + c_2$
  - n and m would be 1 at the best case
- **Worst Case**: Hire all candidates.
  - Cost: $n \cdot c_1 + m \cdot c_2$
  - m would be equal to n because you hire every candidate to find the best one at the worst case.

Let X_i be the indicator random variable for hiring the i^{th} candidate, defined as:

$$X_i = \begin{cases} 1 & \text{if the } i^{th} \text{ candidate is hired} \\ 0 & \text{otherwise} \end{cases}$$

Step 3: Calculate the Expected Number of Hires

Using your notes, we know that the expected number of hires is given by the harmonic series:

$$E[X] = \sum_{i=1}^{n} E[X_i] = \sum_{i=1}^{n} \frac{1}{i} = H_n$$

where H_n is the n^{th} harmonic number, given by:

$$H_n = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n}$$

Step 4: Calculate the Total Expected Cost

Using the formula for the total expected cost from your notes, we have:

$$E[\text{total cost}] = (c_1 \cdot n) + (c_2 \cdot H_n)$$

Step 5: Conclusion

- **Total Expected Cost**:

$$E[\text{total cost}] = (c_1 \cdot n) + (c_2 \cdot H_n)$$

$$E[\text{total cost}] = (c_1 \cdot n) + (\ln(n))$$

Example Two coins throw 20 times. What is the expected number of time you will get both heads and both tails.

Number of times $= \{0, 1, 2, 3, ..., 20\}$ times, we can say 1 represents HH or TT occur 1 time in 20 throws

$X_i =$ indicator random variable, that i represents i^{th} throws

$$X_i = \begin{cases} 1 & \text{if the HH or TT occur for the } i^{th} \text{ throws} \\ 0 & \text{otherwise} \end{cases}$$

$Cost(x),\ x = x_1 + x_2 + x_3 + ... + x_{20}$

apply, Linearity of Expectation Property

$E[x] = E[\sum_{i=1}^{20} x_i]$

$= E[x_1] + E[x_2] + E[x_3] + ... + E[x_{20}]$

$E[x_i] = \sum_x x \cdot P(x_i) = 1 \cdot \frac{1}{2} + 0 \cdot \frac{1}{2}$, When x is 1, then TT or HH

$\sum_{i=1}^{20} \frac{1}{2} = \frac{20}{2} = \sum_{i=1}^{20} E[x_i] = 10$

$P(x)$ is ? Is $P(HH\ or\ TT) = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}$

Chapter 5 Exercises

```
HIRE-ASSISTANT(n)
1 best = 0 // candidate 0 is a least-qualified dummy candidate
2 for i = 1 to n
3   interview candidate i
4   if candidate i is better than candidate best
5     best = i
6     hire candidate i
```

5.1-1 Show that the assumption that you are always able to determine which candidate is best, in line 4 of procedure HIRE-ASSISTANT, implies that you know a total order on the ranks of the candidates.

> We may have been presented the candidates in increasing order of goodness. This would mean that we can apply transitivity to determine our preference between any two candidates

Sure, let's delve into the assumption and its implication in the `HIRE-ASSISTANT` algorithm.

Assumption Understanding

In line 4, it is assumed that you can always identify the best candidate so far. This indicates a clear ranking system among candidates.

Implication of the Assumption

Being able to always pinpoint the best candidate suggests a total order in the ranking of candidates, meaning you can always tell who is better between any two candidates. This is due to the transitive property of the total order.

Example__

If candidates are presented in an increasing order of quality, you can apply transitivity to find your preference between any two candidates. For instance, if candidate 3 is better than 2, and 2 is better than 1, then 3 is better than 1.

This ensures the best candidate up to the current point is always chosen, adhering to a total order based on their rankings.

5.1-2 Describe an implementation of the procedure RANDOM(a, b) that makes calls only to RANDOM(0, 1). What is the expected running time of your procedure, as a function of a and b?

```
Algorithm 1 RANDOM(a,b)
n = ⌈lg(b-a+1)⌉

Initialize an array A of length n while true do
for i = 1 to n do
A[i] = RAN DOM (0, 1)
end for
if A holds the binary representation of one of the numbers in a through b then
return number represented by A end if
end while
```

5.1-3 You wish to implement a program that outputs 0 with probability 1/2 and 1 with probability 1/2. At your disposal is a procedure BIASED-RANDOM that outputs either 0 or 1, but it outputs 1 with some probability p and 0 with probability 1 – p, where 0 < p < 1. You do not know what p is. Give an algorithm that uses BIASED-RANDOM as a subroutine, and returns an unbiased answer, returning 0 with probability 1/2 and 1 with probability 1/2. What is the expected running time of your algorithm as a function of p?

5.2-1 In HIRE-ASSISTANT, assuming that the candidates are presented in a random order, what is the probability that you hire exactly one time? What is the probability that you hire exactly n times?

5.2-2 In HIRE-ASSISTANT, assuming that the candidates are presented in a random order, what is the probability that you hire exactly twice?

5.2-3 Use indicator random variables to compute the expected value of the sum of n dice.

5.2-4 This exercise asks you to (partly) verify that linearity of expectation holds even if the random variables are not independent. Consider two 6-sided dice that are rolled independently. What is the expected value of the sum? Now consider the case where the first die is rolled normally and then the second die is set equal to the value shown on the first die. What is the expected value of the sum? Now consider the case where the first die is rolled normally and the second die is set equal to 7 minus the value of the first die. What is the expected value of the sum?

5.2-5 Use indicator random variables to solve the following problem, which is known as the hat-check problem. Each of n customers gives a hat to a hat-check person at a restaurant. The hat-check person gives the hats back to the customers in a random order. What is the expected number of customers who get back their own hat?

5.2-6 Let A[1 : n] be an array of n distinct numbers. If i < j and A[i] > A[j], then the pair (i, j) is called an inversion of A. (See Problem 2-4 on page 47 for more on inversions.) Suppose that the elements of A form a uniform random permutation of 〈1, 2, … , n〉. Use indicator random variables to compute the expected number of inversions.