The Pigeonhole Principle

Andrew Palacci - CS131 Spring 2021

Presentation Overview

1. The Pigeonhole Principle

Definition and example

Demonstration using Java

2. Generalized Pigeonhole Principle

Definition and example

Demonstration using Java

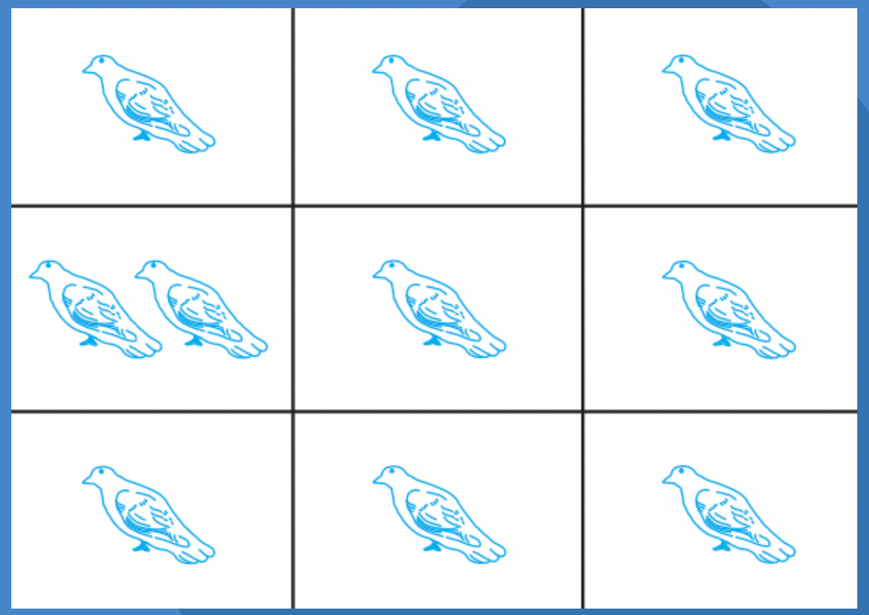
3. Applications of the Pigeonhole Principle

Hair-Counting example

Lossless compression algorithms

What is the Pigeonhole Principle?

- States that "*If k is a positive integer and k + 1 or more objects are placed into k boxes, then there is at least one box containing two or more of the objects.*"
- Example (using pigeons): if 10 pigeons flew into a set of 9 pigeonholes, at least one of the nine holes must contain at least two pigeons.
- This principle can be applied to far more than pigeons, as the rest of the slides will show



Pigeonhole Principle Demonstration

pigeon.java | genPigeon.java

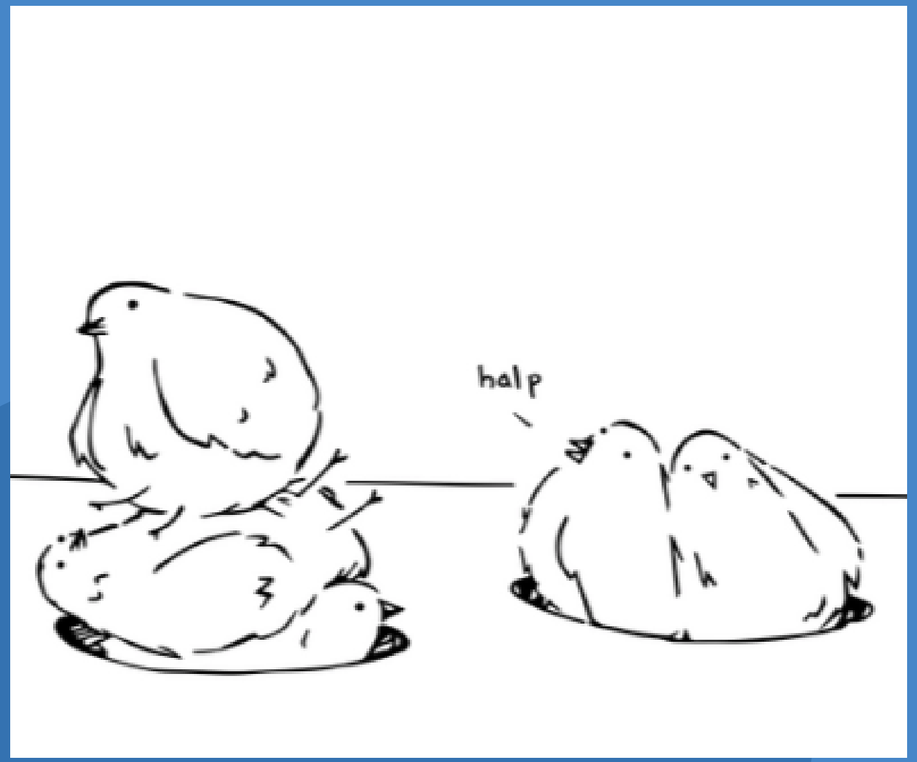
pigeon.java

```java
/*
This program demonstrates the Pigeonhole principle by generating a list of 11 random one-digit integers [0, 9], then showing that
there exists at least one such integer that is repeated at least twice. The program's output shows all of the repeated numbers and
the frequency by which they were repeated, in the form [number] ([frequency]x). Each number is separated by a double space.

If one were to run this program infinitely many times, it would never return a blank output (meaning that there were no repeats). This
can be proven through the pigeonhole principle because we are sorting 11 random integers into 10 boxes (ordered 0-9), so at least one
box must have 2 or more objects (randomly chosen integers) inside of it.
*/

import java.util.*;
import java.math.*;

public class Pigeon
{
  public static void main (String[] args)
  {
    //welcome message
    System.out.println("\nWelcome to the Pigeonhole Principle Demonstration!\n");

    //initializes an array that will contain the random integers
    int[] nums = new int[11];

    //adds random integers to the array nums and prints out the array's contents
    System.out.print("Test Numbers: ");
    for (int i = 0; i < 10; i++)
    {
        nums[i] = (int)Math.floor(Math.random() * 10);
        System.out.print(nums[i] + ", ");
    }
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: zsh

```
Andrew@Androos-Macbook CS131Project %
```

The Generalized Pigeonhole Principle



- States that "*If N objects are placed into k boxes, then there is at least one box containing at least ⌈N／k⌉ objects.*"
- Example: if 55 pigeons fly into the same 9 holes from earlier, there will be at least ⌈55／9⌉ = 7 of them in at least one hole
- This principle is a useful extension of the Pigeonhole Principle because it can be used to make a more specific conclusion when given positive integers N and k

Generalized Pigeonhole Principle Demonstration

pigeon.java | genPigeon.java

genPigeon.java

```java
/*
This program demonstrates the Generalized Pigeonhole principle by taking a list of the most popular male names in the United States
from each first letter, forming a list of 26 names, and then randomly sorting these names into five distinct lists, named List1 - List5.
By the Generalized Pigeonhole principle, one of these lists must contain ⌈26/5⌉ = 6 names in it. The program's output confirms this by
displaying the names contained within each list and then displaying the lists with at least 6 names, as well as the exact amount of names
those lists contain.

If one were to run this program infinitely many times, it would never return a blank output (meaning that there were no lists containing
6 or more names). This can also be proven through the Generalized Pigeonhole principle because we are sorting 26 random names into
5 lists (ordered 1-5), so at least one lists must have 6 or more names in it.
*/

import java.util.*;
import java.math.*;

public class genPigeon
{
  public static void main (String[] args)
  {
    //initializes String array containing the 26 names, num variable for randomization, len for names.length, and the 5 ArrayLists
    String names[] = {"Alexander", "Benjamin", "Carter", "Daniel", "Elijah", "Finn", "Grayson", "Henry", "Isaac",
    "James", "Kayden", "Liam", "Mason", "Noah", "Oliver", "Parker", "Quinn", "Ryan", "Sebastian", "Theodore",
    "Uriel", "Vincent", "William", "Xavier", "Yusuf", "Zachary"};
    int num;
    int len = names.length;
    ArrayList<String> List1 = new ArrayList<String>();
    ArrayList<String> List2 = new ArrayList<String>();
    ArrayList<String> List3 = new ArrayList<String>();
    ArrayList<String> List4 = new ArrayList<String>();
    ArrayList<String> List5 = new ArrayList<String>();
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: zsh

```
Andrew@Androos-Macbook CS131Project %
```

Applications of the Pigeonhole Principle

- Hair-Counting example:
  - The maximum amount of hairs a human head can have is approximately 1,000,000
  - Los Angeles County is home to 10.1 million people, so by the Generalized Pigeonhole Principle, there are at least $\lceil 10{,}100{,}000/1{,}000{,}000 \rceil = 11$ people in Los Angeles County with *exactly* the same amount of hairs on their head
- Lossless compression algorithms:
  - The Pigeonhole Principle allows us to show that it is *impossible* to create a lossless compression algorithm that always reduces a file's size
  - This is because a compression that always reduces file size would result in different files being placed in the same "pigeonhole," and thus having the same compressed result.
  - Since the different files were compressed into the same result, at least one of them must have lost information, showing that a perfectly lossless compression is *not possible*

Works Cited (texts and images)

Hamel, Lutz. “Section 6.2.” , 6 Sept. 2012.

Jin. “Pigeonhole Principle,” *Jineral Knowledge*, 14 Jan. 2014, jineralknowledge.com/pigeonhole-principle/.

Rosen, Kenneth H. *Discrete Mathematics and Its Applications*. New York, Ny, Mcgraw-Hill, 2019.