Alijah Jackson - ajacks11
Darius-Stefan Iavorschi - diavor01

CS 40: Homework 1 - Files of Pix

Problem:

Extracting original image data from corrupted PGM files involves addressing the issues caused by deliberate manipulation of the file structure. Specifically, the header of the PGM file is removed, legitimate rows are interspersed with fake rows, and valid rows are infused with a predictable sequence of non-digit bytes. The goal is to separate the legitimate image rows by identifying and excluding injected rows and to restore the original image content in a "raw" PGM format (P5). To achieve this, the program must process the corrupted file in a single read-through without reopening or skipping within the file, requiring efficient and accurate data structure usage.

Use Cases:

- **Restoring Corrupted PGM Files**: Identifying and retaining valid image rows while discarding injected ones, thereby restoring the original image.
- **Validating Input File Integrity**: Ensuring the corrupted file adheres to the predictable characteristics of the corruption
- **Efficient File Processing**: Handling files with arbitrary-length rows without performance bottleneck.
- **File Conversion**: Converting P2-format PGM files to the P5 format as part of the restoration process, enabling compatibility with modern image tools like Pnmrdr.

Assumptions:

- **Injected Rows Differentiation**: Injected rows and original rows can be identified by the unique infusion sequence of non-digit bytes in legitimate rows. All original rows share the same infusion sequence, while each injected row has a unique sequence.
- **Image Dimensions**: All images are assumed to be at least 2x2 pixels.
- **File Format**: Corrupted images originate from "plain" PGM (P2) files, where each row ends with a newline character (\n). This newline remains unaltered during the corruption process and can be relied upon for row demarcation.
- **Maxval Consistency**: The original images have a consistent Maxval of 255, and this value remains unchanged despite corruption.

Constraints:

- **Performance Requirement**: The entire restoration process, including reading, processing, and outputting the restored image, must complete within 20 seconds for typical input sizes.
- **Memory Management**: Hanson's data structures, including Lists, Tables, and Sequences, must be used where appropriate, with careful consideration to avoid memory leaks. However, the Hanson Array structure is unavailable for this assignment.
- **Single Pass Processing**: The program must read the corrupted file sequentially, without re-opening or seeking within the file, ensuring efficient data handling.
- **Error Handling**: The program must raise Checked Runtime Errors if the file is invalid, unreadable, or lacks expected properties.

Architecture and Implementation:

- For readaline, we will be using a pointer to char called buffer to store the characters. Buffer stops receiving input when finding the endline character.

- For the restoration part, we need 2 data structures: a matrix in numbers matrix_nums (a pointer to pointer to integer), which will represent the final, restored p2 pgm file, and a pointer to char, which will store Hanson's atoms. We prefer this implementation over a pointer to a pointer to char because Atoms allow pointer equality. Since the lines from the original file have been injected with the same sequence of characters, all atoms we store must be equal.

- We will use the readaline function to read the corrupted file line by line. The input will be separated into 2 parts: matrix[i] (a pointer to int, where we store the numbers from the input line) and atoms[i] (where we store anything else). After every line iteration, we check if we found any 2 equal atoms (which should be easy considering atoms allow pointer equality). We repeat the process until we find 2 equal atoms, atoms[x] and atoms[y]. We delete all the lines from both data structures besides x and y.

- Now that we know the correct sequence of characters with which the original lines have been injected, we simply need to check every corrupted line individually and store the numbers in matrix_nums if the sequence matches.

- In the end, matrix_nums should represent the original file, which can safely be converted to a binary format.

Pseudocode for Restoration

```
//func to separate numbers from characters while ( reading ch from the corrupted line):
char* seq_chars = NULL

if ch is digit:
        append ch to matrix_nums[size_matrix][i]
        i++
else:
        append ch to seq_chars

atoms[size_atoms] = Atom_string(seq_chars)
Size_atoms++


//func to check equal sequences for (i: 0 -> size_atoms-1):
   for (j: i+1 -> size_atoms):
      if atoms[i] == atoms[j] found the correct sequence of characters, return [i, j], delete all rows besides i and j

return [-1, -1]
```

Data Structures Used

char* buffer
int** matrix_nums
char* atoms (its elements will be initialized with atoms[x] = Atom_string("sequence_of_injected_characters")
All data structures are allocated dynamically so that they can be resized.

Testing:

- Unreadable File: Input file that cannot be read due to insufficient permissions or a locked state. We expect a checked runtime error to be raised tot thast that the function handles file read open errors
- File Read Interruption: A scenario where file where the reading process is interrupted unexpectedly (e.g., by an external process). We expect a checked runtime error to be raised to verify that the function detects and handles unexpected interruptions during file reading.
- Large Image File: Input a large valid PGM file (e.g., 5000x5000 pixels). We hopefully expect acorrectly restored P5 file in under 20 seconds to ensure the program meets the performance constraint for large inputs.
- Initial read input a file containing a single line longer than 1000 characters. We expect to have initially A runtime error indicating the input line is too long but after adding on after validating it can handle more this would be removed. This is to initially verify error handling for excessively long lines, as specified for partial credit implementations.
- Invalid Argument: Call the restoration command with more than one argument. We expect a checked error to occur and because of an invalid argument call
- Small Corruptiopn: Input a small PGM file (3x3) with only one injected corrupted row. We expect thevalid rows are retained, and the corrupted row is removed in the restored P5 file to Test the ability to handle minimal corruption.