Ece Dilara Aslan
2017400183

CMPE 230 Systems Programming

Project 2

1. Global Variables

hashes: (dictionary)
It stores hashes as keys and list of files/directories as values.
If the valid option is -c or -n, values are list of files/directories that have same hash value and keys are their hashes.
If the valid option is -cn, values are list of files/directories that have same -c and -n hash value and keys are tuples of their -c and -n hashes.
sizes: (dictionary)
It stores the size of files/directories which have same hash value as values and their hashes as keys.
args:
It stores command line arguments, flags and directories if it is given.
duplicates: (list)
It stores duplicate files/directories with/without their sizes, according to -s option is valid or not.
If the -s option is valid, it is a list of tuples. Second index of tuples are lists that stores duplicate files/directories and first index is size of that duplicate files/directories.
If the -s option is not valid, it is a list of lists. Lists are consists of duplicate files/directories.

2. Functions

duplicate():

This function returns the list of duplicate files/directories. It loops through "hashes" and tries to find a list which has more than one element, because duplicate elements have same hash and so they are in the same list. If -s option is valid, it appends tuples of duplicate elements' list and their sizes to "duplicates" list. If -s option is not valid, it appends lists of duplicate elements. Before appending, it sorts the list of duplicates alphabetically. There are two helper functions named myFunc(tuple) and myFunc2(tuple). They will be used for sorting when -s option is valid. After it finds all duplicates, if -s option is valid, it sorts the duplicate sets according to their sizes in descending order and if size of some duplicate sets are equal, it sorts them alphabetically. If -s option is not valid, it sorts the duplicate sets alphabetically again.

traverseFile(directory):

This function is called when -f option is valid. It traverses given directory in BFS order. When it reaches a file, it calculates its hash according to -c, -n or -cn option and it finds the size of the file. After that, it tries to store the hash and size of the current file. If the hash value is already stored, then it checks whether the file is stored before because a directory and its subdirectory can be given as arguments. If not, it stores the hash and path of that file into "hashes". If the hash value is not stored before, then it stores hash and path into "hashes ", hash and size into "sizes".

traverseDirectory(directory):

This function is called when -d option is valid. It traverses given directory in DFS order. When it reaches a file, it calculates its hash according to -c, -n or -cn option. When it reaches a directory, it recursively calls itself. If valid option is -c or -n, hashes of children of current directory appends to "children1". If valid option is -cn, current directory's children's hash value of -c option appends to "children1", hash value of -n option appends to "children2". Also, sizes of children are added to "size". After finding all hashes of current directory's children, we call hashDirectory(directory, children1, children2) function and receive hash of current directory. Then, if the hash value is already stored, it checks whether the file is stored before because a directory and its subdirectory can be given as arguments. If not, it stores the hash and path of that file into "hashes". If the hash value is not stored before, then it stores hash and path into "hashes ", hash and size into "sizes".

hashDirectory(directory,children1,children2):

This function returns the hash value of given directory. First of all, it sorts "children1" and "children2". If the valid option is -n, inserts the hash of directory's name at the beginning of the "children1". If the valid option is -cn, inserts the hash of directory's name at the beginning of the "children2". Then, concatenate all of the hashes in "children1" and "children2". If the valid option is -cn, returns the tuple of hash of concatenated strings in "children1" and "children2". If the valid option is -c or -n, returns the hash of concatenated strings in "children1".

parse():

This function parses and returns the command line arguments. "parser" has a mutually exclusive group of -f and -d arguments. Also, it has -c, -n and -s optional arguments and dirs argument.

3. General Procedure

First, program parses the command line arguments. If -f option is valid, it calls traverseFile(directory) function, if -d option is valid, it calls traverseDirectory(directory) function and the given parameter changes with respect to "args.dirs". If "args.dirs" is an empty list, then the parameter is current directory, if not, it calls appropriate function for each given directory and give them as parameter. After arranging "hashes" and "sizes", program calls duplicate() in order to find duplicates. Then, if -s option is valid, prints the duplicate files/directories and their sizes as groups, if not, prints only the duplicate files/directories as groups.