

Write a program that compresses files and folders into a set of compressed files such that of each compressed file doesn't exceed a maximum size. The same program can be used for decompressing the files that it has generated earlier. The output of the decompression should be identical to the original input of the compression process.

When compression is needed, the program takes 3 parameters:

1. Path to Input directory. This directory will contain some files and folders.
2. Path to Output directory. This is where the program should write the compressed files to.
3. Maximum compressed size per file expressed in MB.

When decompression is needed, the program takes 2 parameters:

1. Path to Input directory. This directory contains the compressed files that were generated by the same program.
2. Path to Output directory. This is where the program should put the decompressed files & folders.

Consider the following when writing the code:

1. Some files may be greater than the JVM memory.
2. Some input files (even when compressed) may be greater than the maximum size allowed for output files.
3. You can use zip for compression algorithm (using JDK's implementation don't use a third party library for that) but design the program to allow support different compression algorithms in the future.
4. When compressing, generate as few files as possible. You are not required to generate the absolute minimum number of compressed files but doing so would be a plus.
5. <<Bonus>> Make the compression process run in parallel to speed up the compression for a single directory.

Write your production-quality code in Java or Scala. Submit your solution by using the link that was included in the email.