**Programming process steps in aver\_per\_sim\_1\_haplo.py**

1. Import necessary Python modules including csv, os, time, concurrent.futures, multiprocessing, and tempfile.

2. Prevent system sleep on Windows using Windows API calls.

3. Define a function process\_roll\_forward\_and\_average\_streaming that computes time-averaged population statistics across simulation runs up to the fixation or last generation.

4. Define a specialized function process\_roll\_forward\_and\_average\_streaming\_fix\_A that processes only runs where allele A fixed (freq\_A ≈ 1.0), always including generation 0.

5. Define a specialized function process\_roll\_forward\_and\_average\_streaming\_fix\_a that processes only runs where allele a fixed (freq\_a ≈ 1.0), always including generation 0.

6. Define a worker function process\_one\_sim\_nr that reads per-generation data for a single SimNr from the input file.

7. In process\_one\_sim\_nr, parse and filter input rows belonging to the target SimNr using early exit for sorted data.

8. Convert field values to appropriate types (int or float) and skip malformed rows.

9. If valid data exists, compute averaged time-series for heterozygosity loss using process\_roll\_forward\_and\_average\_streaming.

10. Write the averaged results for heterozygosity to a temporary file named by SimNr.

11. Compute and write separate averaged time-series for runs where allele A fixed to a second temporary file.

12. Compute and write separate averaged time-series for runs where allele a fixed to a third temporary file.

13. Return the SimNr and success status to the main process.

14. Define a helper function get\_sim\_nrs that scans the input file to extract all unique SimNr values while checking for proper sort order.

15. In the main function, record the start time and verify that the input file "out\_1\_haplo\_per\_gen.txt" exists.

16. Initialize the primary output file "A\_haplo\_AVE\_vals\_per\_gen\_1H.txt" with a header row.

17. Scan the input file to collect and sort all SimNr values.

18. Create a temporary directory to store intermediate per-SimNr result files.

19. Prepare a list of arguments for parallel processing, one per SimNr.

20. Set the multiprocessing start method to 'spawn' for compatibility.

21. Launch parallel workers using ProcessPoolExecutor to process each SimNr independently.

22. As each worker completes, report its success or failure and track progress.

23. After all workers finish, concatenate all temporary heterozygosity result files into the primary output file.

24. Initialize and write the header for the allele-A fixation output file "haplo\_AVE\_A\_per\_gen\_1H.txt".

25. Append all temporary A-fixation result files into this output file.

26. Initialize and write the header for the allele-a fixation output file "haplo\_AVE\_a\_\_per\_gen\_1H.txt".

27. Append all temporary a-fixation result files into this output file.

28. Print the names of the three output files.

29. Print the total program execution time.