Coding task: BOM checker

We perform mechanical design of our devices using a 3D CAD (SolidWorks). Such a mechanical design is a collection of part files (`*.sldprt`) organized in subdirectories and typically contains a few thousand files.

In addition to the CAD part files, we write down all part files in a "part list" file called a bill of materials (BOM).

The 3D CAD part files are easy to change (e.g., simply by moving the view angle and accidentally hitting Ctrl-S) and this poses a problem because it is difficult to tell those innocuous changes (e.g., view angle changes) from potentially critical errors (e.g., accidentally dragging and changing an important dimension). To mitigate the chance of breaking the design, this BOM contains a SHA256 hash of each part file (`*.sldprt` file) so that we can detect unintentional file changes by calculating the SHA256 hash of actual files and comparing them to the BOM.

Task

Your task is to implement a small Python tool that checks if the CAD part files (`*.sldprt`) are accidentally modified or otherwise inconsistent with the BOM. This tool will compare the actual part files (`*.sldprt` files) in the folder to the BOM and detects following types of errors.

- Modification of part files (SHA256 hash inconsistent to what is recorded in the BOM)
- Missing part part files (`*.sldprt` file specified in the BOM but missing from the folder)
- Unidentified part files (`*.sldprt` file present in the folder but missing from the BOM)
- Duplication of part files (multiple `*.sldprt` files with the same name in different folders)

To perform this task, Python standard library packages `hashlib` and `csv` will likely be relevant.

Requirements

- The code will be executed from command line as `python bomchecker.py "path/to/root"`.
  - Python 3.8 will be used.
  - `path/to/root` is the top-level folder that contains CAD part files. The code shall only check contents of this folder.
- The code shall only use the Python Standard Library.
- If a part file is specified in the BOM but not present under `path/to/root`, its filename shall be printed in stdout.
- If a part file is present under `path/to/root` but is missing from the BOM, its path shall be printed to stdout.
- If SHA256 is specified in the BOM and it does not match SHA256 calculated from the real file, its path shall be printed to stdout.
  - If SHA256 is not specified in the BOM (blank), the corresponding part file (`*.sldprt`) shall not be reported as modified.
- If multiple part files with the same name (case-insensitive) are present under `path/to/root` folder, their paths shall be printed to stdout.
- If none of the problems above is detected, output `OK` to stdout.
- Any files that do not have the extension of `*.sldprt` shall be ignored.
- A BOM csv file (`bom.csv`) is present at `path/to/root/bom.csv`.
- You can assume the BOM csv file is properly formatted:
  - The first row is a header row.
  - Following the header row, it contains 1 to 10000 rows.
  - Each row contains a unique and valid part file name.
  - Each row contains a blank or valid SHA256 hash. Some files may be contain a blank SHA256, indicating no file hash check for that file.
  - The BOM csv file contains only ASCII characters.
  - The BOM csv file does not contain any double quotes (`"`), commas (`,`), and new lines (`\n`) within a cell.

Output format

The detected errors shall be printed to stdout in the following format.

```
MODIFIED: test2\sub1\sub1-2\part A.sldprt
MISSING: part B.sldprt
UNIDENTIFIED: test2\sub2\part not in BOM.sldprt
DUPLICATE: test2\sub1\sub1-1\part A.sldprt
```

If none of those errors are detected, print `OK` to stdout.

Evaluation criteria

- Correctness of the program
- Code organization and cleanliness

Example

See `sample1` folder. By running the code, it should detect 6 errors. The output may appear as below (the ordering of lines may vary).

```
> python bomchecker.py sample1
MODIFIED: sample1\sub1\sub1-2\part A.sldprt
MODIFIED: sample1\sub1\sub1-2\part D.sldprt
MISSING: part B.sldprt
UNIDENTIFIED: sample1\sub2\part not in BOM.sldprt
DUPLICATE: sample1\sub1\sub1-1\part A.sldprt
DUPLICATE: sample1\sub1\sub1-2\part A.sldprt
```