```
const fs = require("fs");
const path = require("path");

function sortImportsInFile(filePath) {
    let code = fs.readFileSync(filePath, "utf8");
    const lines = code.split("\n");

    const useClientIndex = lines.findIndex(line => line.trim() === '"use
client";');
    const importIndices = [];

    lines.forEach((line, idx) => {
        if (/^import .+;$/.test(line.trim())) {
            importIndices.push(idx);
        }
    });

    const importLines = importIndices.map(i => lines[i]);

    const processedLines = importLines.map(line => {
        const match = line.match(/^import\s+([^\{\s]+)?\s*(,\s*)?\{([^}]+)\}/);
        if (match) {
            const defaultImport = match[1] ? match[1].trim() : null;
            const namedImports = match[3]
                .split(",")
                .map(i => i.trim())
                .sort((a, b) => a.localeCompare(b))
                .join(", ");

            if (defaultImport) {
                return line.replace(/\{[^}]+\}/, `{ ${namedImports} }`);
            } else {
                return `import { ${namedImports} } from ${line.split("from")
[1].trim()}`;
            }
        }
        return line;
    });

    processedLines.sort((a, b) => a.length - b.length);

    importIndices.forEach((idx, i) => {
        lines[idx] = processedLines[i];
    });

    if (useClientIndex !== -1 && useClientIndex !== 0) {
        const [useClientLine] = lines.splice(useClientIndex, 1);
        lines.unshift(useClientLine);
    }

    fs.writeFileSync(filePath, lines.join("\n"), "utf8");
}

// **Check for file path argument**
const fileArg = process.argv[2];
if (!fileArg) {
    console.error("❌ Please provide a file path. Example: node scripts/sort.js
src/pages/Home.jsx");
    process.exit(1);
}

const resolvedPath = path.resolve(fileArg);
sortImportsInFile(resolvedPath);
```

```
console.log(`✅ Imports sorted in ${fileArg}`);
```