

# CSC 370 Concepts of Programming Languages - Spring 2023

## Homework #3

Due: Wednesday, 02/15/2023 (before class)

Submission method: Cardinal Learn

**\*\* Combine all of your answers into a single PDF file. Note: Failure to do so will result in 0**

### Dirsort

- Sean can't remember where he put an important file on his computer. He desperately needs the file tonight to finish his homework, since he procrastinated all day by playing computer games. Unfortunately, his computer has gotten fried from all the gaming, and can no longer perform automatic searches for files. He therefore has to look for the file manually.
- Since he tends not to use subdirectories too much, he knows his best bet is to look for the file first in the root directory ("/"), and next in directories only one level deep (for example, "/Games/"), and so forth, checking the "deepest" directories last. His computer still has enough fire power to run some simple programs, so he has asked you to code for him a simple algorithm to sort all of the directories on his computer.
- The algorithm should take a **String[] dirs** as an input and should sort dirs first by directory depth, and then lexicographically for each depth. So "/d/e/" comes before "/a/b/c/", but not before "/c/d/". Also, "/a/bc", comes before "/ab/c", since "a" comes before "ab" in lexicographical order.

### Constraints

For a directory to exist, it's superdirectory does not necessarily need to exist. For example, you can have "/usr/admin/" without having "/usr/" or even "/".

- dirs will contain between 1 and 50 elements, inclusive.
- each element of dirs will be of length 1 to 50, inclusive.
- each element of dirs will contain only lowercase letters [a-z], inclusive, and the slash ('/') character.
- each element of dirs will begin with a slash, end with a slash, and not have double slashes anywhere.

## Examples

a) {"/","/usr","/usr/local","/usr/local/bin","/games","/games/snake","/homework","/temp/downloads/"}

**Returns:**

```
{ "/", "/games/", "/homework/", "/usr/",  
  "/games/snake/",  
    "/temp/downloads/", "/usr/local/", "/usr/local/bin/" }
```

Notice that "/temp/downloads/" can exist even without "/temp/". The sorted String[] is:

```
{ "/",  
  "/games/",  
  "/homework/",  
  "/usr/",  
  "/games/snake/",  
  "/temp/downloads/",  
  "/usr/local/",  
  "/usr/local/bin/" }
```

b) {"/usr","/usr/local","/bin","/usr/local/bin","/usr/bin","/bin/local","/bin/local/"}

**Returns:**

```
{ "/bin/", "/usr/", "/bin/local/", "/bin/local/",  
  "/usr/bin/",  
    "/usr/local/", "/usr/local/bin/" }
```

c) {"/","/a","/b","/c","/d","/e","/f","/g/"}

**Returns:**

```
{ "/", "/a/", "/b/", "/c/", "/d/", "/e/", "/f/", "/g/" }
```

d) {"/","/a","/b","/c","/d","/e","/f","/g","/a/a","/b/g/c","/g/f/"}

**Returns:**

```
{ "/", "/a/", "/b/", "/c/", "/d/", "/e/", "/f/", "/g/",  
  "/a/a/", "/g/f/", "/b/g/c/" }
```

e) {"/a/b/c/d/e/f/g/h/i/j/k/l/m/n","/o/p/q/r/s/t/u/v/w/x/y/z/"}

**Returns:**

```

        { "/o/p/q/r/s/t/u/v/w/x/y/z/",
          "/a/b/c/d/e/f/g/h/i/j/k/l/m/n/"
        }

```

f) { "/a/b/", "/ab/cd/", "/c/d/", "/a/b/c/", "/ab/c/d/", "/a/bc/d/",  
"/a/b/cd/" }

**Returns:**

```

{ "/a/b/", "/ab/cd/", "/c/d/", "/a/b/c/", "/a/b/cd/",
  "/a/bc/d/", "/ab/c/d/" }

```

**Grading Questions:**

P1. Implement your class "**Dirsort**" using a programming language of your choice (i.e. Python, Java, C++, etc.). When completed, i) upload your code to your CSC370 repository, ii) take a screenshot of your code and include it in your submission.

P2. Explain if your "**Dirsort**" program was implemented with any form of data abstraction.

P3. Explain if your "**Dirsort**" program was implemented with any form of control abstraction.