Craving For Packages

Due date: May 29, 2015 at 23:55 CET

> *This is an individual assignment. You may discuss it with others, but your formulations, your code, and all the required material must be written on your own.*

Your company is developing USI OS, the next big thing in the tech world, and you have been assigned to the implementation of the package management system.

In USI OS every program is distributed as a package, and each package may require another package as a *dependency*. Each package is composed of the following elements:

1. a unique name
2. a dependency list: a list of packages that must be installed before the current package

Sometimes, a package may offer an alternative between two dependencies, meaning that only one of the two packages is required.

The names of the packages and the dependencies are defined in a file with the following format:

```
package_name: dependency1, dependency2, dependency3a | dependency3b
```

where:

- *package_ name* is the name of the package
- the name of the package is separated by the list of the dependencies by the colon character '**:**'
- each dependency is separated by a comma '**,**'
- when two dependencies are separated by the pipe character '|' only one of the two packages needs to be installed to satisfy the dependency

Example

Given the file with the following contents:

```
package1:
package2: package1
package3: package1, package2
package4: package1 | package3
package5: package1, package2 | package3
```

we can assert that:

- *package1* has no dependencies
- *package2* depends on *package1*
- *package3* depends on *package1* **and** *package2*
- *package4* depends on *package1* **or** *package3*
- *package5* depends on *package1* **and** (*package2* **or** *package3*)

Problem 1 — Load and Search (30 points)

Write a program that reads from the standard input the name of a file containing a list of packages and their dependencies in the format described above, then create a data structure to manage the loaded data.

Implement the function SEARCH, that accepts the name of a package as a parameter and answers *true* if your data structure contains the named package and *false* if it does not.

Problem 2 — Dependencies (70 points)

Implement the function DEPENDENCIES that accepts the name of a package as a parameter, and prints a minimal set of packages required to install the package.

In the context of the example above, your algorithm should give the following answers:

dependencies(package1) prints nothing

dependencies(package2) prints *package1*

dependencies(package5) prints *package1*, *package2*

Implementation and Submission

Write your program in Python, version 3. You may not use any library function other than to read data from the input. You may not use any *pre-defined* data structure other than basic arrays or lists.

You must make sure that your program can run directly from the terminal.

You must submit a single tar.gz or zip archive containing only:

1. all the necessary sources to run your program;
2. a README.pdf or README.txt;
3. a data file containing an example set of packages to load in your program.

Do not include any other file or directory. You may use the IDE of your choice, but do not include any project files and folders.

The README file should contain a brief description of your implementation, possibly a list of limitations or errors you are aware of but that you were not able to fix, clear references to any and all external material you might have used – including discussions with or help from other students – and the instructions to run your program. Also, you have to discuss the complexity of your algorithms (build the data structure, search and dependencies).

Name your archive file following this format: assignment04g-*lastname-firstname*.tar.gz (or .zip). Submit the tar.gz or zip archive through the *iCorsi* platform.