**CSI 403 – Project #3 – Who is Smartest?**

Goal:

* Write a list of people ordered so that no one appears in the list before anyone he or she is less smart than.

Problem:

* Provide a RESTful service which accepts as a POST of JSON a list relationships [A, B] such that A is smarter than B.
* Return in JSON the list sorted by smartness, with the smartest person first.
* Example input: (“Einstein is smarter than Feynmann”, “Feynmann is smarter than Gell-Mann”, etc.)

{ “inList” : [

{ “smarter” : [ “Einstein”, “Feynmann” ] },

{ “smarter” : [ “Feynmann”, “Gell-Mann” ] },

{ “smarter” : [ “Gell-Mann”, “Thorne” ] },

{ “smarter” : [ “Einstein”, “Lorentz” ] },

{ “smarter” : [ “Lorentz”, “Planck” ] },

{ “smarter” : [ “Hilbert”, “Noether” ] },

{ “smarter” : [ “Poincare ”, “Noether” ] }

] }

Example output:

{ “outList” : [ “Einstein”, “Feynmann”, “Gell-Mann”, “Thorne”, “Lorentz”, “Planck”,

“Hilbert”, “Poincare”, “Noether” ] }

* Erroneous input (e.g. malformed JSON) should be handled gracefully.
* For reference see Cormen Ch. 22 regarding topological sorting.

Deliverables:

Submit to the Blackboard by the due date:

* An HTTP URL to a RESTful service which must remain up and running 24/7 until grading is complete. Graders will invoke your service with a tool such as curl or Postman at a time of their choosing.
* A ZIP file containing your source code, written in any language you choose.

{ "inList" : [ { "smarter" : [ "Einstein", "Feynmann" ] }, { "smarter" : [ "Feynmann", "Gell-Mann" ] }, { "smarter" : [ "Gell-Mann", "Thorne" ] }, { "smarter" : [ "Einstein", "Lorentz" ] }, { "smarter" : [ "Lorentz", "Planck" ] }, { "smarter" : [ "Hilbert", "Noether" ] }, { "smarter" : [ "Poincare", "Noether" ] } ] }