Rita Yousif

Week 6: Final Project Module 3

The goal of the project is to match four interns to four projects using their skills to the project required skills. The intern cannot work on multiple projects at the same time because as an Intern myself it is hard to manage multiple projects, which is what motivated me to make this project. The way to do this would be using a greedy algorithm in order to make this match where the skills of the projects would be required for the intern and matched accordingly based on the order of skills listed in the projects skills. The greedy algorithm class has a function called greedy with the parameters of internSkillsMap and projectSkills. This returns a hashmap keys as intern names and values as the projects. Therefore, the main logic contains the definition of a stack, which stores an entry of the internSkillsMap. Go through the stack while it is not empty and pop the element (internWithSkill) each time and loop over the array of project skills . Then, check if the skillsWithProjectMap has a project with that skill and the intern does not have a project assigned to them previously, and if so, then assign it to the intern. We are using greedy because we are iterating through the project skills and it chooses to assign project for the intern who is the first one available by their skill and it does by the order of skills listed in the array of project Skills.

Main class:

```
import java.util.*;

import java.time.LocalDate;

//Rita Yousif

public class Main {

   public static void main(String[] args) {

       double begin=System.nanoTime();

       //Hash Map creation named projectMap where it creates a key of project
and a string array , which is the

       // value of the hash map that are for the skills needed for the project.

       HashMap<Project, String[]> projectMap = new HashMap<>();

       //this is the declaration of the projects with their dates of starting
```

```
days and endings days

      Project firstProject= new Project(

              LocalDate.of(2026,5,12),

              LocalDate.of(2026,5,22), "firstProject");

       Project secondProject= new Project(

               LocalDate.of(2026,5,12),

               LocalDate.of(2026,5,22), "secondProject");

       Project thirdProject= new Project(

               LocalDate.of(2026,5,12),

               LocalDate.of(2026,5,22), "thirdProject");

       Project fourthProject= new Project(

               LocalDate.of(2026,5,12),

               LocalDate.of(2026,5,22), "fourthProject");

       //this is the requirements of skills for each of the projects

       projectMap.put(firstProject, new String[]{"Arabic", "English"});

       projectMap.put(secondProject, new String[]{"English"});

       projectMap.put(thirdProject, new String[]{"Spanish"});

       projectMap.put(fourthProject, new String[]{"French"});

       //this for loop would iterate through the projectMap that makes a return
of a set of keys and values from

       //the map where the key is the object of the project adn values are
array of the skills needed

       // which links each project to its needed skill.

       for (Map.Entry<Project, String[]> entry : projectMap.entrySet()) {

           System.out.println("Key: " + entry.getKey().projectName+

                   ", Value: " + Arrays.toString(entry.getValue()));

       }
```

```
        //this hash map is made for the intern skills where it contains the key
as the name of the intern

        //and the value as the set of string of skills that the intern acquires
to know the skills of each intern

        //this would be used to give the intern project using their skills.

        HashMap<String, Set<String>> internSkillsMap = new HashMap<>();

        // this would add to the internSkillsMap the skills of the intern to the
intern using their name

        internSkillsMap.put("internA", new HashSet<>(Set.of("English",
"Spanish")));

        internSkillsMap.put("InternB", new HashSet<>(Set.of("Arabic",
"Chaldean")));

        internSkillsMap.put("internC", new HashSet<>(Set.of("Spanish",
"French")));

        internSkillsMap.put("internD", new HashSet<>(Set.of("Turkish",
"Protugese")));

        //This is another hash map that is made to make a connection between the
skills of the intern and the project

        //that needs that skill.

        HashMap<String, Project> skillsWithProjectsMap = new HashMap<>();

        //This is initialization of projectSkills array in the form of strings
where it contains skills for the project

        //I need for all projects where it would give projects to interns using
these skills

        String projectSkills[] = {"Arabic", "English", "Chaldean", "Spanish",
"French"};

        //this would see if the skill has been given to a project

        Boolean given = false;

        //Basically, this would iterate throughout the projectSkills for each
skill, then check projects in map to

        //know if there is a skill needed for that project and then check if
there is project that needs a skill to add
```

```
        //that to the skillsWithProjectMap in the form of an entry

        //then when a project has a matching skill that is needed for the
projectSkill, it gives that project and breaks

        //in order to not give multiple projects using that skill.

        //Then, each of the skills with the project it has get to be printed

        for (String skill : projectSkills) {

            for (Map.Entry<Project, String[]> entry : projectMap.entrySet()) {

                for (String projectSkill : entry.getValue()) {

                    if (projectSkill.equals(skill)) {

                        skillsWithProjectsMap.put(projectSkill, entry.getKey());

                        given = true;

                        break;

                    }

                    if(given) {

                        break;

                    }

                }

            }

        }

        for (Map.Entry<String, Project> entry :
skillsWithProjectsMap.entrySet()) {

            System.out.println(entry.getKey() + " →  [ " +
entry.getValue().projectName + " ]" );



        }

        //this is using the greedy algorithm where the alg is being used as an
instance

        // it is calling the greedy algoirthm function
```

```
        GreedyAlgorithm alg = new GreedyAlgorithm();

        HashMap<String, Project> project1Map = alg.greedy(internSkillsMap,
skillsWithProjectsMap, projectSkills);



        //this iterating through the project1Map and printing the intern name
with the project

        for (Map.Entry<String, Project> entry : project1Map.entrySet()) {

            System.out.println(entry.getKey() + " → " +
entry.getValue().projectName + " → " + entry.getValue().startTime

                    + " → " + entry.getValue().endTime);

        }

        double stop=System.nanoTime();

        double time=stop-begin;



        System.out.println("The runtime of the code is " + time + " in
nanoseconds");
```

Greedy Algorithm Class:

```
import java.time.LocalDate;

import java.util.HashMap;

import java.util.Map;

import java.util.Stack;

import java.util.*;

//Rita Yousif

public class GreedyAlgorithm {

      // The greedy algorithm class has a function called greedy with the
parameters of internSkillsMap
```

```
      // and projectSkills. This returns a hashmap keys as intern names and
values as the projects.

      // Therefore, the main logic contains the definition of a stack, which
stores an entry of the internSkillsMap.

      // Go through the stack while it is not empty and pop the element
(internWithSkill) each time and loop over

      // the array of project skills . Then, check if the skillsWithProjectMap
has a project with that skill and

      // the intern does not have a project assigned to them previously, and if
so, then assign it to the intern.

       public HashMap greedy(HashMap<String, Set<String>> internSkillsMap,
HashMap<String, Project> skillsWithProjectsMap, String [] projectSkills) {

               HashMap<String, Project> internWithProjectMap = new HashMap<>();

               Stack<Map.Entry<String, Set<String>>> stack = new Stack<>();

               for (Map.Entry<String, Set<String>> entry :
internSkillsMap.entrySet()) {

                       stack.push(entry);

               }

               while (!stack.isEmpty()) {

                       Map.Entry<String, Set<String>> internWithSkill =
stack.pop();

                       for (String skill : projectSkills) {

                               if (skillsWithProjectsMap.get(skill) != null &&
internWithSkill.getValue().contains(skill)) {


internWithProjectMap.put(internWithSkill.getKey(),
skillsWithProjectsMap.get(skill));

                                       break;

                               }

                       }

               }
```

```
                return internWithProjectMap;

        }

}
```

Project Class:

```
import java.time.LocalDate;
//Rita Yousif
class Project {
   LocalDate startTime;
   LocalDate endTime;
   String projectName;


//this is the duration of the project with start time and end time
   public Project(LocalDate start, LocalDate end, String project) {
       this.startTime = start;
       this.endTime = end;
       this.projectName = project;
   }


}
```

Test Plan and Result:
Test: The way I tested this out was by having overlapping times, overlapping skills, intern A with skill at second place of the projectSkills[], and not assigning a matching skill to a project to an intern.
Result: All of the projects had the same starting times and ending times that were printed next to the interns and the project they got assigned. Intern C had two skills that matched two different projects and it gave third project to it because Spanish was listed before French was. French was assigned to the fourth project, so in this case it did not give the fourth project to intern C because it already was working on the third project. Intern A was skilled at English and English was listed second place in projectSkills[], so it got the second project. Intern D has no matching skill to project, so it is not assigned to one. Intern C and intern A both had Spanish as a skill they share, but the skill was given to the first available intern, so it gave it to intern C since A got second project already. Each skill was linked to a project. They all get one project, so it passed the tests.

Expected: Each intern received a project based on their order in the projectSkills[] because of the use of the greedy algorithm.

Actual: As shown for intern C received the third project rather than fourth because of the greedy algorithm choosing to give project with the skill of spanish instead of French since it was listed first, which was the language in the third project project.

Runtime: The preliminary run time of the greedy algorithm is 2.3036875E7 in nanoseconds. This was done using the nanoTime() method in the main class around the entire code to get the whole runtime of the project using begin time and end time of the project.

Enhancement of Project Quality In Future: In the future, this project can be enhanced by not allowing the project to be assigned solely on intern skills, but also on the time they are available. This means if there was a starting date and an end date on a project then those dates would be compared first. If the both start and end at the same time, then the project would be assigned based on the skill order listed that has been matched. Right now, the fourth project is not assigned to intern C, which is a goal, but also for future, I would want it to be listed as a future assignment to work on.

https://github.com/ryousif1694/yousif_rita_final