

CSE 321 – Introduction to Algorithm Design

Homework 03

Deadline: 29 November 23:55

PLEASE DO NOT FORGET TO READ THE NOTES CAREFULLY

THE NOTES ARE UPDATED, AGAIN

1. Watch the film “Devrim Arabalari”(2008). Write your thoughts about the film using an office program. It will at least occupy half of the A4 paper. **(10 points)**
2. **(36 points)** You have become the head of computer engineering department. You want to assign assistantship of each course to a R.A. (Research Assistant). To do this, you must consider skills of each R.A., some are good at Algorithms while some others may be expert at Operating Systems. According to his/her skills, an assistant may do the assistantship of a course more easily than the other ones.

Your input will be a 2-dimensional Python list of integers(*table*) with size $n \times r$ where n is number of courses and r is number of R.A.s (assume $r \geq n$ always). Inside the table, the element at `table[i][j]` has the meaning of assistant number i must spend `table[i][j]` hours for course j per week. Design and implement a **brute force** algorithm in Python 3 to find the optimal solution of minimizing the total time spent for course assistantship for the whole department. If number of courses (n) is less than the number of R.A.'s (r), unoccupied R.A.'s will do the other department tasks, each of which has a cost of 6 hours per week. Your function will return 2 things: first a 1-dimensional Python list of integers with size r , `list[i]` will hold the number of the course that R.A. i will be its assistant and -1 if it is doing another department stuff. The second return value is a single integer denoting the minimum total time spent per week.

Your code will be tested using the code block given below. It will be also tested by using many different inputs, so make sure your solution is general. Moreover, explain your algorithm **in pdf file** and do its complexity analysis.

Your function will be implemented in a separate python file named **assistantship** **_StudentID.py**. Hint:

```
# Courses 0 1 2
inputTable = [[5, 8, 7], # R.A. 0
              [8, 12, 7], # R.A. 1
              [4, 8, 5]] # R.A. 2

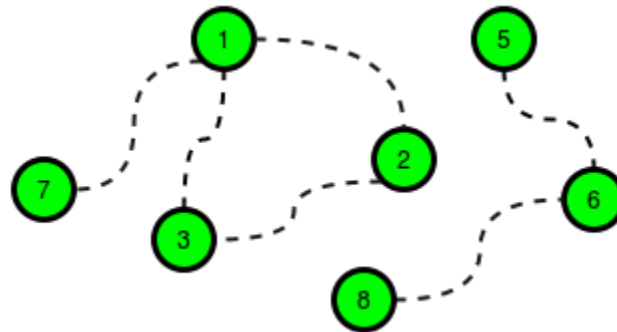
asst, time = findOptimalAssistantship(inputTable)
print(asst)
print(time)
Output:
[1, 2, 0] #R.A. 0 is assigned to course 1 etc.
19 # Minimum total time
```

3. **(36 points)** The rector of Gebze Technical University believes that every department of the university should have access to a computer lab. Unfortunately, GTU was hit by a flood that destroyed all of its computer labs and obstructed its roads! As you are the greatest programmer of GTU, the rector wants your help to repair the roads and build some new computer labs efficiently.

GTU has n departments numbered from 1 to n . The departments are connected by m bidirectional roads. Students of a department have access to a computer lab if:

- Their department contains a computer lab.
- They can travel by roads from their department to another department containing a computer lab.

The following figure is a sample map of GTU where the dotted lines denote obstructed roads:

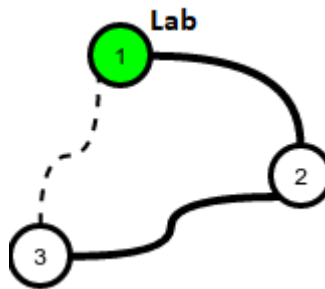


The cost of repairing any road is x liras, and the cost to build a computer lab in any city is y liras. Design an algorithm (not brute force) and implement in Python 3 to find the minimum cost of making computer labs accessible to all the students. Inputs of your function will be x (the cost to build a computer lab), y (the cost to repair a road), `mapOfGtu` (graph that represents GTU). The graph will be a python dictionary (a data structure in python, hash map), its keys will be integers denoting vertex number and its values will be python sets (another data structure in python) that holds other vertex numbers which are connected to the key vertex. Therefore, the graph is represented as an adjacency list.

Your function will be implemented in a separate python file named `labify_StudentID.py`. Implement and use DFS or BFS in your algorithm as a separate Python 3 function. That will help you a lot. Also explain your algorithm in pdf file and do the worst case analysis.

Example 1:

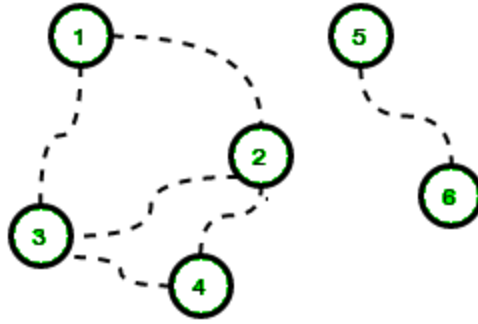
```
mapOfGTU = {  
    1 : set([2,3]),  
    2 : set([1,3]),  
    3 : set([1,2])  
} # graph is initialized as dictionary  
  
minCost = findMinimumCostToLabifyGTU(2,1,mapOfGTU)  
print(minCost) # Output will be 4
```



Cost of building a lab is 2 and repairing a road is 1. There are 3 departments and 3 roads. The optimal solution is to build a lab at any of the departments (cost 2) and repair 2 roads (cost $1 \times 2 = 2$). Hence, the minimum total cost is 4. Solution is represented by the graph above.

Example 2:

```
mapOfGTU = {  
    1 : set([2,3]),  
    2 : set([1,3,4]),  
    3 : set([1,2,4]),  
    4 : set([3,2]),  
    5 : set([6]),  
    6 : set([5])  
} # graph is initialized as dictionary  
  
minCost = findMinimumCostToLabifyGTU(5,2,mapOfGTU)  
print(minCost) # Output will be 18
```



Input graph is represented above. The cost of building a lab is 5 and repairing a road is 2. There are 6 departments and 6 roads. The optimal solution can be to build a lab at 1 and repair 3 roads to reach 2,3, and 4. Also we need to build a lab at 5 or 6 and repair the road between them. Consequently, the minimum total cost is $2*4 + 5*2 = 18$.

Furthermore, do not forget that if the cost of building a lab is less than repairing a road, then the solution is build a lab at every department.

4. **(18 points)** Apply Insertion sort and Shell sort to list 12, 34, 54, 2, 3 . Show each step in the pdf. What are the advantages of shell sort compared to insertion sort? Discuss the differences between them.

IMPORTANT NOTES

1. You will write your answers in Microsoft Word, Libre Office or similar office program. No photos, no handwritten papers.
2. Codes will be written using Python 3 (not Python 2). Do not use any additional python libraries. Use only sys library(import sys).
3. Homeworks will be submitted in ZIP format. The file hierarchy will be this:
CSE321_HW3_StudentID.zip
| → CSE321_HW3_ANS_StudentID.pdf
| → assistantship_StudentID.py
| → labify_StudentID.py
4. Use homework question forum on moodle if you have any questions about homework.
5. Cheating will be punished. Taking any code from internet is also forbidden. (Grade will be -100)
6. No late submissions.