UNIVERSITY OF MALAYA
BACHELOR OF COMPUTER SCIENCE / BACHELOR OF INFORMATION TECHNOLOGY
ACADEMIC SESSION 2019/2020: SEMESTER 2

WIA1002/WIB1002 Data Structure

Assignment

GROUP FORMATION

1. Form a group of not more than 5 members. All the members must be from the same tutorial group.
2. Each of the groups will be given one of the listed projects randomly.
3. Every group member must contribute to the project, including certain amount of coding.
4. The role played by and contribution of each member to the project must be included in the managerial report.

SUBMISSION

Your group has to submit the softcopy of the items listed below:

1. A technical report explaining the assigned task, the requirements of the task, the approach taken to solve the task, a detail description of your solution (including the flowchart, modules, etc.), sample snapshot of your program output.
2. A managerial report explaining the formation of the group, role and assigned work for each of the members, the project timeline, the problems faced in accomplishing this assignment and your solutions, and other issues encountered.
3. The complete source code.

For softcopy, the group leader will submit them in one .zip file to Spectrum. The name of the zip file should be *TutorialGroupNumber-GroupLeaderName*.zip.

IMPORTANT DATES

- Submission: (week 13th) **12:00pm on Friday, 19th June 2020**
- Viva/demo: (week 14th) during tutorial & lab

MARKING SCHEME: (20 MARKS)

1. Source code/Program. (16 marks)
    a. Meeting the basic task requirements, i.e. identify the requirements of the task and produce a workable solution - 10 marks
    b. Extra features/functionalities that are not included in the task specification - 6 marks.
2. Reports (4 marks)

WARNING

Plagiarism is prohibited, should there be any evidence found,
it will result in a heavy penalty on your assignment grades!

Good luck and have fun!

Topic 1: Jumpy Grof

Problem Statement:

You work in a wildlife research centre in Australia. Your friend Grof is very interested in how kangaroos move around in the wild area. He told you that kangaroos are herbivores and only eat certain foods such as carrots, apples, etc. Studies also shown that male kangaroos tend to go to places with more female kangaroos for certain purpose. Grof also said that kangaroos are intelligent, and their behaviour is similar to human. They will form colonies when the group is large enough and move in groups. Since you are an excellent programmer, Grof asked you to help build a simulator that simulates kangaroos' behaviour in the wildlife.

Kangaroo Simulator:

Imagine a wildlife area with plenty of spaces and resources. The area is inhabited by a group of kangaroos each starting at random places. Different area has different amount of resources, some has more, some has none. Male kangaroos move around by jumping from one point to another while female kangaroos will not move. However, certain places will be more difficult to travel to, and some kangaroos do not have the ability to do so. Hence, the simulator should start with a map (randomly generated/user input). A map consists of different points where kangaroos can stay and use the resources there. Each kangaroo has different needs and start and random points in the map. Kangaroos will jump to another point if the current point's resource is depleting. If there are extra foods in the point, the kangaroo will keep the extras in its pouch. When the number of kangaroos in a place exceed a certain threshold, they will form a colony and stay at the place forever. Kangaroo can join the colony if the resource there is enough for everybody (including foods in pouch). The simulation ends when no kangaroo is moving around anymore.

Point:

Each point represents area that kangaroos can stay in the map. Point have resources/food for available and each point can only house a limited number of kangaroos. Points on the map are connected to each other however, each path (connection) has a high obstacle. If two points are connected, kangaroo can travel freely between the points as long as it can jump over the obstacle and the point is not full yet. For example, Point A is connected to point B. Point B has more food/resource than in point A but the path AB is blocked by a wall of height 10. So, all kangaroos that are able to jump 10 high will travel to point B for more food.

Kangaroo:

Kangaroo can be either male or female. Female kangaroo will not move around the map but it will still store foods into its pouch. Each kangaroo will have a pouch that can hold a limited amount of food. Whenever a kangaroo reaches a new point, it will first eat the food to restore energy used to travel to the new point and keep the remaining food in its pouch (food needed = height of obstacle + 0.5 * food in pouch). The food needed to restore energy is proportional to the obstacle's height and the amount of food carried in pouch. Before a kangaroo decides to leave the current area, it will first check all nearby points that it can jumps to and move to the point that will results in more food and more female kangaroos. For example, point A with 0 food is connected to point B with obstacle height 5 and point C

with obstacle height 8. Point B has 10 food and point C has 15 food. Kangaroo X at point A is carrying 8 foods. It will jump to point C as after reaching there and eating 9 (15 - (8 + 0.5*8)) food to restore energy, point C will still have 3 foods remaining. However, if point A currently has more than 3 foods, the kangaroo will not jump to point C. If the remaining food is equal, the kangaroo will decide based on the number of female kangaroos.

COLONY:

A colony is formed, when the number of kangaroos (including female) in the point exceeds a certain threshold. The threshold will be set by user. When a colony is formed, kangaroos will stay at the point and will not move away. All foods carried by the kangaroos will be shared with each other. When new kangaroo arrives at the colony, it will be forced to join the colony if it is carrying food enough for everyone as a gift and the limit of the point is not reached. For example, kangaroo Y carrying 3 foods try to join a colony of 5 kangaroos will be rejected and kangaroo Y will not be able to pass through the colonized point.

INPUT:

The map can be either randomly generated or input by user.

If input by user, the first line of input will be an integer *n*, the number of points that is in the network. The following lines will describe the *n* instances of points, each separated by a blank line. In each instance, the first line contains four integers, *a*, *f*, *l* and *m*, with *a* representing the ID of the point, *f* representing the number of food available in the point, *l* represents how many kangaroos the point can fit while *m* represents the number of path connected to the point. The next *m* lines will contain 2 integers, *A* and *h*. *A* is the ID of the other point which is connected to the current point while *h* is the height of the obstacle in the path.

The second input contains information of each kangaroo available in the map. First line of the input will be an integer *x*, the number of kangaroos. The following *x* lines will contain 3 variables, *i, s,* and *p. i* is the point where the current kangaroo starts at, *s* is the gender of the kangaroo (M is male, F is female) while *p* is the amount of food that can be stored in its pouch.

Final line of input will indicate the threshold amount of kangaroo to form a colony.

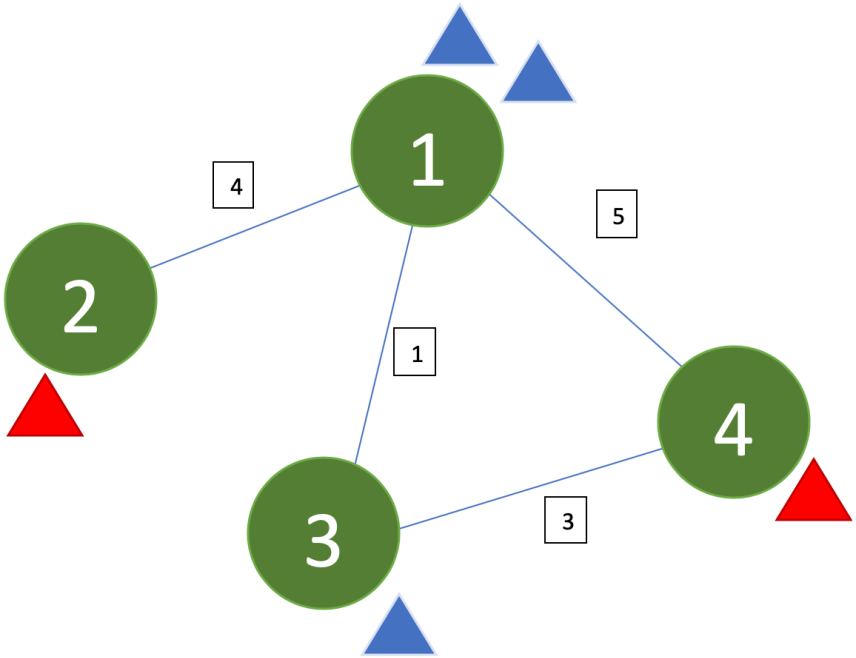
Feel free to modify the input format as you see fit. You may change to console prompt but do not leave out any information.

OUTPUT:

The output should consist of *z* lines with *z* representing the sum of number of colonies and number of remaining kangaroos. For each colony, print out the total number of kangaroos in that colony. For each remaining kangaroo, print out the location of the kangaroo, gender of kangaroo and how much food is in its pouch.

Sample Input:

```
4
1 16 10 3
2 4
3 1
4 5

2 5 3 0

3 10 8 1
4 3

4 11 6 0

5
1 M 6
2 F 5
4 F 3
1 M 10
3 M 6

3
```



Sample Output:

```
Number of colonies: 1
3
Number of remaining kangaroos: 2
2 F 5
3 M 8
```

Explanation: All kangaroo including females will start by collecting food into their pouch. The amount of food available at point 1-4 will be, 0, 0, 4 and 8 respectively. Kangaroos will take turn moving according to the sequence of input. First kangaroo will move from point 1 to point 4 because point 2 has 0 food while moving to point 3 and point 4 will result in no increase in food but point 4 has more female kangaroos. The movement will use up all 8 food in point 4. The fourth kangaroo will then move from point 1 to point 3 because point 3 has the most food. The movement will cost all 4 foods in point 3 and 2 foods from the pouch. Now all point has no more food. The kangaroos will now only move to adjacent point if there are more female kangaroo and it has enough food for the movement. The fifth kangaroo will move from point 3 to point 4 and use up all food in the pouch. The three kangaroos in point 4 will now form the first colony. The fourth kangaroo now tries to move to point 4 but will be rejected because the movement will cost 7 foods and the remaining 1 food will not be enough to be gifted to the 3 members of the colony. At this moment, all kangaroo cannot move, and the output is printed.

ASSUMPTION:

1. Each path is bi-directional, meaning the kangaroo can travel back and forth between the points
2. The number of points in the map will always be less than 20.
3. When the kangaroo is idle, it will not consume any food. Food is only required when moving around
4. There can only be at most one path between any two points.

EXTRA FEATURES:

1. GUI

- Visualize the maps. Your program may show the whole map with all the points and kangaroos. Your program may also animate the jumping movement of each kangaroos.

2. Unidirectional path

- Assume each path has only one direction. This means that now there can be at most two paths connecting any two points, one back and one forth. Each direction will have different obstacle height.

3. Optimization

- In the current simulation, each kangaroo only has knowledge about their nearby points. Hence, their movement is limited. If the kangaroo has the knowledge of the whole map, it can then choose to go to a point that will result in lesser food but after jumping through a few points, it will end up with more foods and female kangaroos.

4. Food regeneration

- In real world, food resource will regenerate. Hence, after certain amount of time, food can regenerate to its starting amount. (If you wish to implement this, you should show how food regeneration had affected the final output)

5. Any idea

- Add any idea that can help Grof in his research.

TOPIC 2: FRIEND ZONE

INTRODUCTION:

The Anti-Couple Party (ACP) has come up with a new plan to destroy the market of couples. Many couples were born in online dating applications such as Tinder and Tantan. As a member of ACP, I will never let this continue. Now listen carefully, here is the **Friend Zone** plan...

PLAN #0: FAKE THE DATING APP

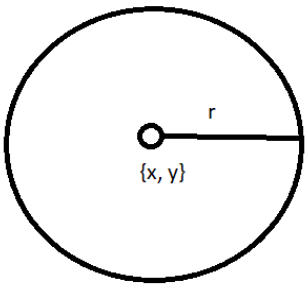
You need to make a fake dating app. Although you need to use Java to implement the app, you can choose any suitable platforms you want such as CLI, Mobile App or Web App. You can use a text file or other Database to store the data. I would recommend using SQL as using a text file just gives us pretty bad performance and your program wouldn't be able to cope with millions of users.

PLAN #1: GENERAL AUTHENTICATION

To get as many people on board with this fake app, your authentication must be **special.** It has to be able to let users sign up with any existing dating app accounts. For example, a user with an account **TinderUser** can use the existing account to sign up a new account with the dating app. Similarly, a new user with an account TantanUser can use the existing account to sign up a new **FriendZoneUser** account. Simply said, you need to create an **AdapterClass** to convert other User classes to FriendZoneUser class, ask for more information if you need them to create a class. (Read Adapter Pattern in Java to get more context.)

PLAN #2: FIND USERS NEAR ME

This is a VERY IMPORTANT feature, please be aware of the following explanations. Every user will have their own x and y coordinate, indicating their geographical location.



Above is an example of a user. You need to return a list of users which lies inside the circle around the selected user. Let's say in the first search, r is set to 100 meters. All the users that are within the range of 100 meter from the selected user will be put into a result list and returned. If no user is found, the search range will be increased, let's say r will be set to 200 meter now. The operation will be continued as long as there is no user found and stopped when reaching a MAX r value that you searched. Take note that this is a **circle** not a **rectangle**!

Plan #3: Find The Same Interests

Now you have a list of users nearby. You need to sort those users by the similarity in terms of interests. You can set your own weightage; the easiest way is to increment score by 1 when there is a similar interest between two users. Justify your resultant sorted list when presenting your feature. By the way… let the user choose the gender of users they are interested in to avoid any issues.

Plan #4: Troll Message Feature

Now the main course comes in. You need to prepare a basic messaging feature. This is the core troll feature for Friend Zone. Your program has a dictionary of words and their antonyms. When a user sends a message to another user, the message content will be converted to antonyms if possible. For example, LittleSugar sent a message to Guy X:

LittleSugar: You are so handsome!!!

The message will then be converted to:

LittleSugar: You are so ugly!!!

Plan #5: Substitution Encryption

Same as any chat systems in the market, we need the messages to be encrypted when storing in our database. No worries, we will just use a very simple encryption method: The Substitution method. You will have a dictionary that maps a character into another character when encrypting the message content, for example:

{'a': 'z', 'o': 'x', … }

Following the above dictionary, the sentence " You are so ugly!!!" will be encrypted to "Yxu zre sx ugly!!!" when stored in the database. You will need to decrypt the messages when retrieving the messages from the database too. You need to handle upper cases, lower cases, numbers and perhaps some common special characters.

Plan #6: Integrate All Modules

Use your own logical way and integrate all the above-mentioned features into a program. We're ready to roll out this amazing application.

Conclusion: Membership of ACP

Implement the features efficiently. Make your codes work well when there are millions of users in the database. By the way, you are free to do more than this to get marks for extra features. Example of extra features is better encryption methods. Any questions regarding the assignment, or to apply for membership of Anti-Couple Party (ACP), contact dsfriendzone001@gmail.com (serious). CC your lecturer in the email to align better :D

TOPIC 3: *EL-CORONA*

INTRODUCTION:

As the Covid-19 outbreak has since been declared as a global pandemic, it is vital for all of us to stay vigilant at all times at keeping our personal hygiene. If an admitted patient is tested positive for Covid-19, subsequent measures by the authorities in performing contact tracing and isolation must take place in order to contain the virus.

During the contact tracing process, the admitted patient will be interviewed, through phone, to gather information on his whereabouts over the past two weeks or more.

"Where have you been to?"
"From when to when?"
"How did you get from point A to point B?"
"Who else was around?"

The officials will then look up for who else was at the same location at that time and clustering similar cases together (hint: suitable data structures like bag or graph) for easier identification.

While humans are humans, it is difficult for us to have a perfect memory. During the contact tracing process, there will eventually be some missing points in between, which causes ambiguity. Incomplete data poses a challenge to the contact tracing process, potentially leaving out critical information, which may then lead to sporadic cases, indicating a community outbreak.

DESCRIPTION:

There will be two parts of the assignment:

1. Human activity simulator, and,
2. Contact tracer

HUMAN ACTIVITY SIMULATOR

In order to simulate human activity, you will need to define a map, which can be an urban city ***or*** rural countryside. They will differ in terms of population (density), age distribution, the types of occupations, education level, and etc.

For instance, in the rural countryside, it will probably have a small and aged population. People who live there could be farmers, fishermen, small business owners, etc. Besides, in terms of the infrastructure and services available, there would be a hospital with a smaller capacity as compared to an urban city, as well as fewer road connectivity.

Your task is to come up with a program that simulates daily human activity within the region that you have defined. *The entire simulation is expected to be output into multiple log files.* The simulation should be as long as possible (at least 14 days) until there is **ONE** patient confirmed for Covid-19, be it an imported case from other cities or contracted by the patient while traveling.

Note: Once there is a case confirmed, contact tracing will commence immediately, which will be the second part of this assignment.

CONTACT TRACER

Once there is someone who has confirmed to be Covid-19 positive, contact tracing will have to be carried out **immediately** in a bid to rapidly contain the virus from spreading before it is too late. *The log outputs from the human activity simulator will act as the input for the contact tracer.*

Interviews will be conducted in order to gather as much information from the patient as possible, ***However, human memory is not perfect.*** He/ she may forget some important events in between, potentially leaving out some potential contact points!

Your task is to write a program to help the Health Ministry to list out all recent close contacts/ locations for that particular patient. Depending on the quality of the information provided, you should provide an analysis by comparing the information provided by the patient versus the ground truth of your simulation. As an example, the statistics for how many patients who are infected but are not able to be traced should be listed out.

CONSIDERATIONS:

Things that you may consider in your simulation (not limited to):

1. The population can be divided into multiple categories according to their age to correspond to their forgetfulness. For instance, Children (0-30%), Adults (5-50%), Senior Citizens (10-70%)
2. The population can also be divided according to their education level and civic awareness.
3. Various occupations may have different probabilities of getting infected due to the nature of their respective jobs.
4. Every individual would have different levels of immunity, which affects the probability of infection.
5. Every individual would have different daily schedules! They can go to many places in a day. Just like you.

Note: Your simulation should be as realistic as possible. The more attention you pay, the more merits you deserve to be credited for your assignment.

SUGGESTED DATA STRUCTURES:

Here are some data structures that you may find useful:

1. Graphs (human connections, locations)
2. Tree/ Trie (contact tracer)
3. LinkedList (activity log)

MINIMAL REQUIREMENTS:

For your contact tracer, given infected human ID, infection date, infection probability threshold, output a list of possible infected human IDs, and print the list as a tree, for each depth, sorting by [dates (latest first), infection probability (highest first)]. Let's limit the depth of the tree to 5.

To make it simple, assume the infection process is unidirectional only, which is you can see in the output sample below, just to calculate the infection probability without any external factor. For example, 6789 will not be infected from any places except contacted by 5678. The deeper the tree, the lesser the infection probability.

However, if 6789 is a friend of 4567 and 6789 first contacted with 4567 as well, then he has to be a direct child of 4567 instead of 5678. See the example below:

// contact with children at the same date
```
4567 20/01/14 0.9
  6789 20/01/15 0.81
  5678 20/01/15 0.81
```

// 4567 contact with 6789 first
```
4567 20/01/14 0.9
  6789 20/01/15 0.81
    5678 20/01/16 0.729
```

// 4567 contact with 5678 first
```
4567 20/01/14 0.9
  5678 20/01/15 0.81
    6789 20/01/16 0.729
```

Input: **humanID dateOfInfection(YY/MM/DD)**
```
1234 20/01/12
```

Output: **humanID firstDateOfContactWithPossibleInfection infectionProbability**
```
1234 20/01/12 1.0
   3456 20/01/13 0.9
     4567 20/01/14 0.81
     5678 20/01/15 0.81
          6789 20/01/15 0.729
          7890 20/01/17 0.729
```

TDLR; Simple Requirements:

1. Random a list of human IDs, with their respective friends (children).
2. Human relationships are complicated, a friend of a child can be a friend of a parent.
3. Display a list of human IDs in tree format.
4. Display a list of possible infected human IDs in tree format given an input of human ID.
5. Display the tree with at least depth of 5.
6. For **infectionProbability**, *you can decide the rule of infection probability,* e.g. for each depth, the decay rate will be 0.9, parent with 1.0, child with of 1*0.9 = 0.9, child of child is 0.9*0.9=0.81.

   Instead of probability, you could also look at the R0 metric (Basic Reproduction Number) to simulate how fast the infection spreads. R0 for COVID-19 is 1.4-3.9, meaning if uncontrolled, a person can spread to 1.4 to 3.9 persons. Simple calculation can be referred over here.

7. For the **firstDateOfContactWithPossibleInfection,** you can random the activity of a human, which is when he contacted his parents, and when he contacted his children, if the child is contacted 14 days before he is infected, then the child is safe and can be excluded from the tree.

GRADING CRITERIA:

ROOKIE (FOR EVERYONE!)

By completing the above-mentioned modules, you will only be entitled to get **12 MARKS** at maximum, subject to the:

1. Attention to detail given in performing the simulations.
    a. You will need to show how your simulations are carried out.
    b. Show proof, e.g., in the form of logs.
2. Quality of the contact tracer visualization and analysis

Note: You will need to justify your choices made accordingly.

Veterans (For the brave ones!)

"Being brave means doing what you are afraid to do." -- Unknown

For your assignment to be considered for **20 MARKS**, in addition to the requirements above, we have something extra for you.

Real-time Simulation and Monitoring

You will still need to perform human activity simulation of a city. However, instead of performing contact tracing in an "offline" manner, in the earlier section, while your simulation is ongoing, you are required to perform **real-time monitoring**. The simulation and monitoring should be ongoing as long as your program is active.

You may refer to the following examples:

1. Singapore's Ministry of Health Dashboard on COVID-19
    a. https://go.gov.sg/covid-19-dashboard
2. Malaysia's COVID-19 Tracker
    a. https://www.outbreak.my/map
    b. https://newslab.malaysiakini.com/covid-19/en

In terms of grading, the more complex your simulation and the more details you have on your monitoring dashboard, the closer you are to getting full **20 Marks**! ;)

Additional Considerations:

What else can you consider in your simulation?

1. How about increased connectivities to the city?
    a. Airports/ Railway stations
2. Multiple cities? How about a whole country?
3. Letting your program to accept input in real-time, in order to implement control measures (as the role of the government). You may refer to examples of:
    a. Singapore (Strict quarantine/ immigration rules)
    b. Malaysia (Movement Control Order)
    c. Hubei Province, China
4. If there are government control measures, what if, there is a small portion of "naughty" citizens who do not follow the orders? Will it affect the infection/ mortality rate of the virus?
5. In terms of hospital capacity, what if there are insufficient resources?

You may add extra scenarios at your own discretion in making the simulations more realistic. There are no restrictions. Your imagination/ ability is the limit for how this assignment will be!

Hint: If you have (3), (4) implemented, by tweaking the parameters, it would be really awesome to see how the infection/ mortality rate would progress in real-time. That could easily earn a full 20 mark.