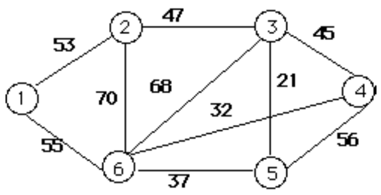
Data Structures and Algorithms
INFO 6205
Homework 10
Due: April 6, 2019

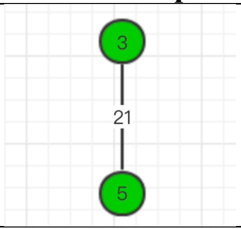
Put all your java, compiled class files and documentation files into a zip file Homework10.zip and submit it via the dropbox on the blackboard before the END of due date. Put your name on all .java files. There will be a short Quiz on this homework.

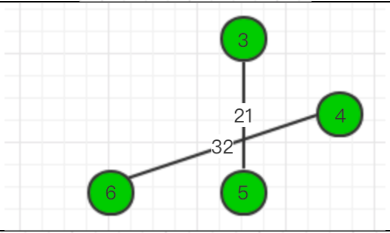
1. Solve the Minimum Spanning Tree for the following Graph,

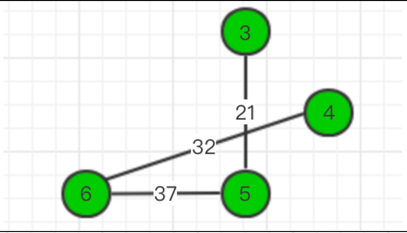


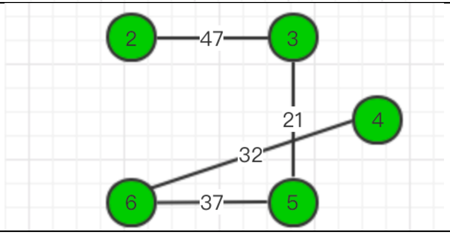
a) Kruskal's algorithm step-by step

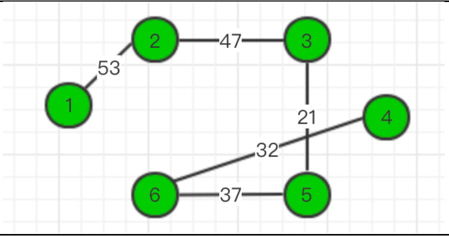
Show Step	Step Detail

	Select edge 3-5
	S = {3-5(21)}
	mstSet = {3, 5}

	Select edge 4-6
	S = {3-5(21), 4-6(32)}
	mstSet = {3, 5, 4, 6}

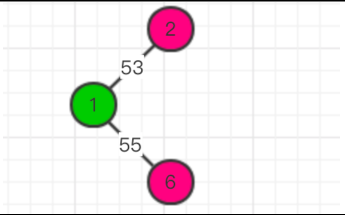
	Select edge 5-6
	S = {3-5(21), 4-6(32), 5-6(37)}
	mstSet = {3, 5, 4, 6}

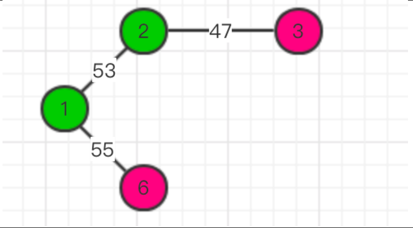
	Select edge 2-3
	S = {3-5(21), 4-6(32), 5-6(37), 2-3(47)}
	mstSet = {3, 5, 4, 6, 2}

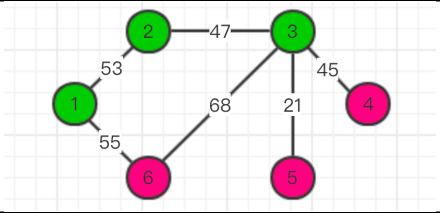
	Select edge 1-2
	S = {3-5(21), 4-6(32), 5-6(37), 2-3(47), 1-2(53)}
	mstSet = {3, 5, 4, 6, 2, 1}

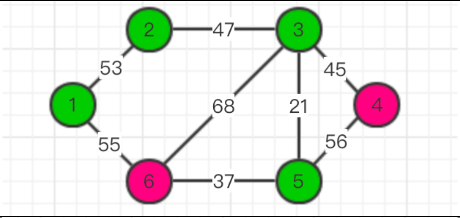
b) Prim's Algorithm step-by-step

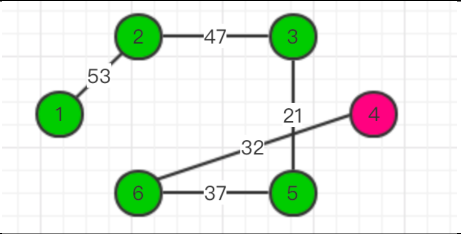
Show Step	Step Detail

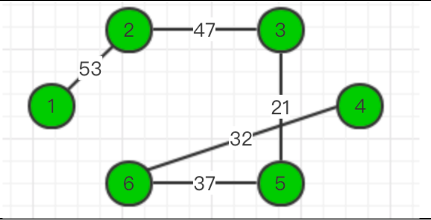
	Select the top vertex 1
	S = {1(0)}
	U = {2(53), 3(INF), 4(INF), 5(INF), 6(55)}
	mstSet = {1}

	Select the next vertex 2
	S = {1(0), 2(53)}
	U = {3(47), 4(INF), 5(INF), 6(55)}
	mstSet = {1, 2}

 2
47
3
53
45
1
68
21
4
55
6
5
	Select the next vertex 3
	S = {1(0), 2(53), 3(47)}
	U = {4(45), 5(21), 6(55)}
	mstSet = {1, 2, 3}

 2
47
3
53
45
1
68
21
4
55
56
6
37
5
	Select the next vertex 5
	S = {1(0), 2(53), 3(47), 5(21)}
	U = {4(45), 6(37)}
	mstSet = {1, 2, 3, 5}

	Select the next vertex 6
	S = {1(0), 2(53), 3(47), 5(21), 6(37)}
	U = {4(32)}
	mstSet = {1, 2, 3, 5, 6}

	Select the next vertex 4
	S = {1(0), 2(53), 3(47), 5(21), 6(37), 4(32)}
	U = {}
	mstSet = {1, 2, 3, 5, 6, 4}

c) Write Java code for (a) and (b) and compile

d) Compare space and time complexity between two algorithms

	Prim’s Algorithm	Kruskal’s algorithm
Time complexity	O（v^2）	O（v^2）
Space complexity	O（v^2）	O（v^2）

e) Is following code Prim’s or Kruskal’s algorithm, explain?

```
1) Make a queue (Q) with all the vertices of G (V);
2) For each member of Q set the priority to INFINITY;
3) Only for the starting vertex (s) set the priority to 0;
4) The parent of (s) should be NULL;
5) While Q isn’t empty
6) Get the minimum from Q – let’s say (u); (priority queue);
7) For each adjacent vertex to (v) to (u)
8) If (v) is in Q and weight of (u, v) < priority of (v) then
9) The parent of (v) is set to be (u)
10)The priority of (v) is the weight of (u, v)
```

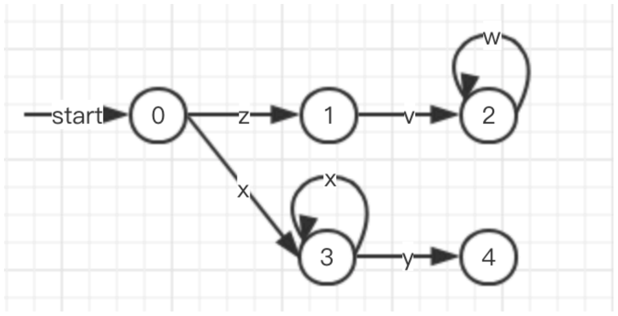
I think this is Prim’s. Because for Prim’s,

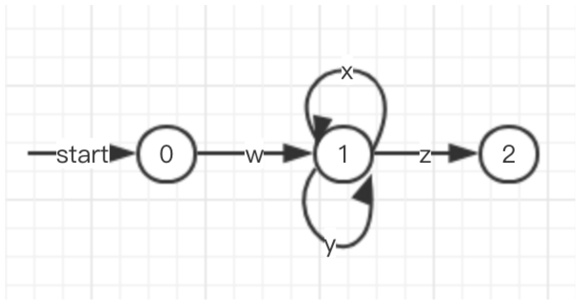
1) Create a set mstSet that keeps track of vertices already included in MST.
2) Assign a key value to all vertices in the input graph. Initialize all key values as INFINITE. Assign key value as 0 for the first vertex so that it is picked first.
3) While mstSet doesn’t include all vertices
4) Pick a vertex u which is not there in mstSet and has minimum key value.
5) Include u to mstSet.
6) Update key value of all adjacent vertices of u. To update the key values, iterate through all adjacent vertices. For every adjacent vertex v, if weight of edge u v is less than the previous key value of v, update the key Value as weight of u v

2. For the following Regular Expression (RE) Input Strings

a) Convert each RE to DFA

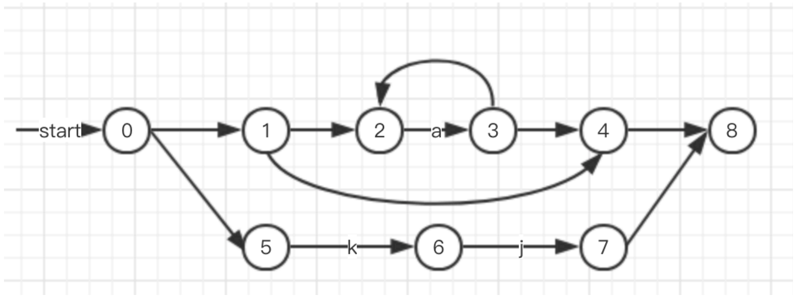
zvw*|x+y   w(x|y)*z

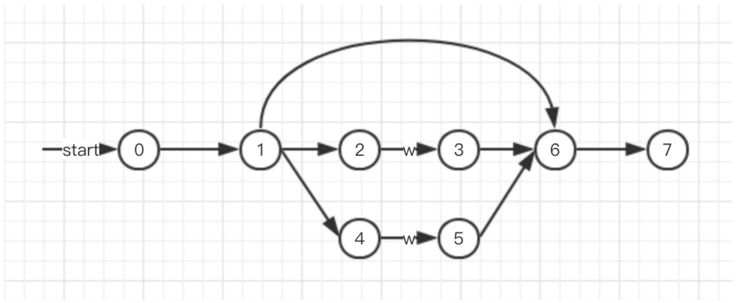




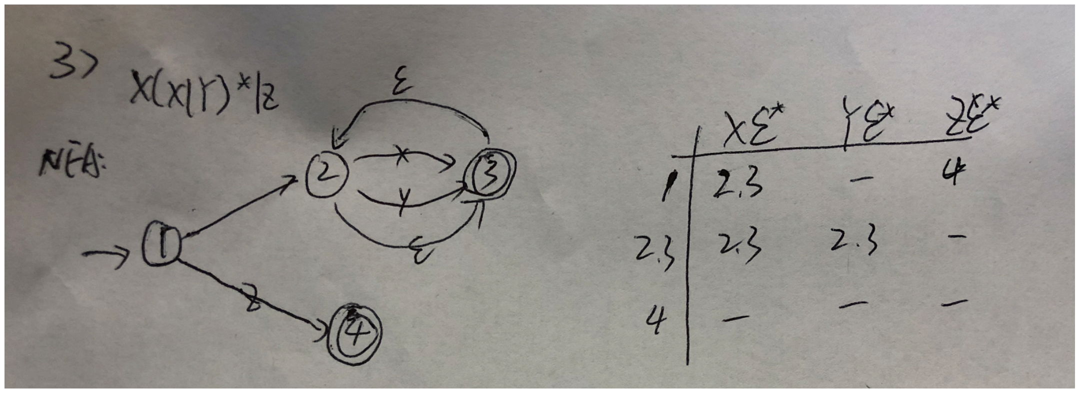
b) convert RE to NFA

a*|kj   (w|x)*

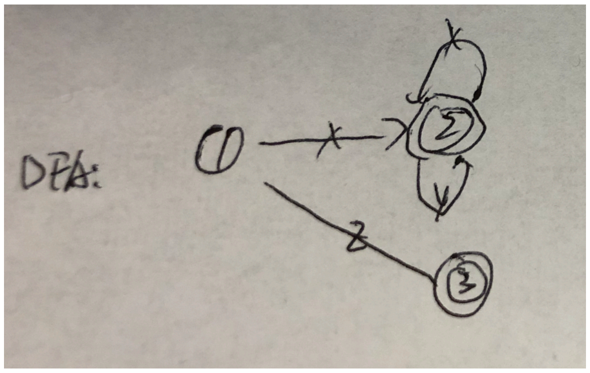




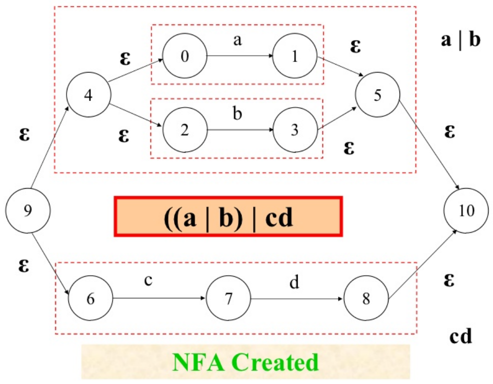
c) Convert X(X|Y)*|Z to NFA and then to DFA

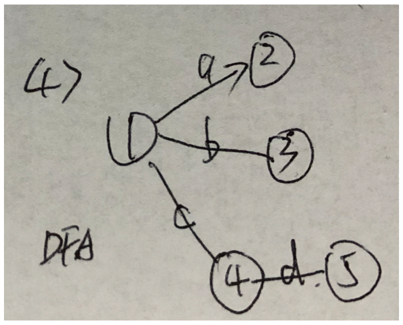


	Xε*	Yε*	Zε*
1	2,3	–	4
2,3	2,3	2,3	–
4	–	–	–



d) convert the following NFA to DFA and then to RE





RE: (a|b)cd

	a	b	c	d
9	1, 5, 10	3, 5, 10	8	-
1, 5, 10	-	-	-	-
3, 5, 10	-	-	-	-
7	-	-	-	8, 10
8, 10	-	-	-	-

3. Read this article on Genetic Algorithm:

https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3

a) What are the steps of GA described in the article?

There are 5 steps of GA.

1) Initial population:

An individual is characterized by a set of parameters (variables) known as Genes. Genes are joined into a string to form a Chromosome (solution). In a genetic algorithm, the set of genes of an individual is represented using a string, in terms of an alphabet.

2) Fitness function:

The fitness function determines how fit an individual is (the ability of an individual to compete with other individuals). It gives a fitness score to each individual.

3) Selection:

The idea of selection phase is to select the fittest individuals and let them pass their genes to the next generation. Two pairs of individuals (parents) are selected based on their fitness scores. Individuals with high fitness have more chance to be selected for reproduction.

4) Crossover:

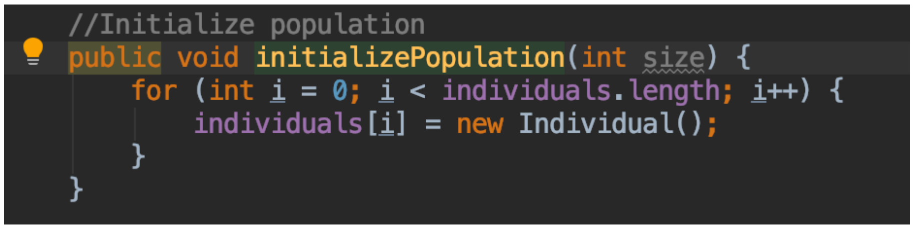
Crossover is the most significant phase in a genetic algorithm. For each pair of parents to be mated, a crossover point is chosen at random from within the genes.

5) Mutation:

In certain new offspring formed, some of their genes can be subjected to a mutation with a low random probability.

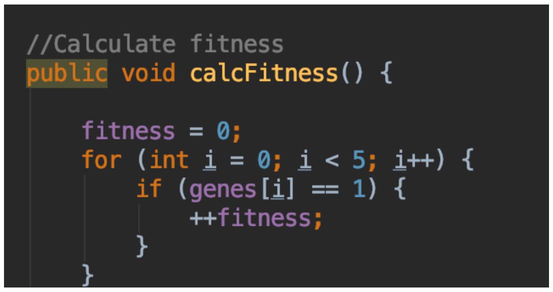
b) Read the example Java code as how it relates to steps in (a)

1) Initial population:



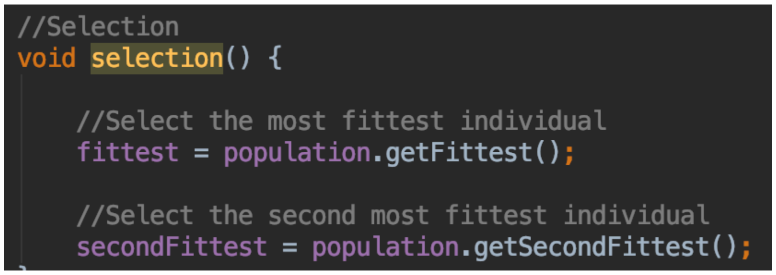
```
//Initialize population
public void initializePopulation(int size) {
    for (int i = 0; i < individuals.length; i++) {
        individuals[i] = new Individual();
    }
}
```

2) Fitness function:



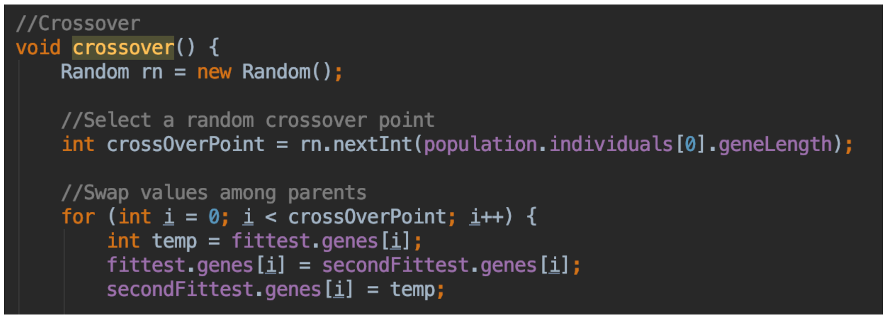
```
//Calculate fitness
public void calcFitness() {

    fitness = 0;
    for (int i = 0; i < 5; i++) {
        if (genes[i] == 1) {
            ++fitness;
        }
    }
```

3) Selection:



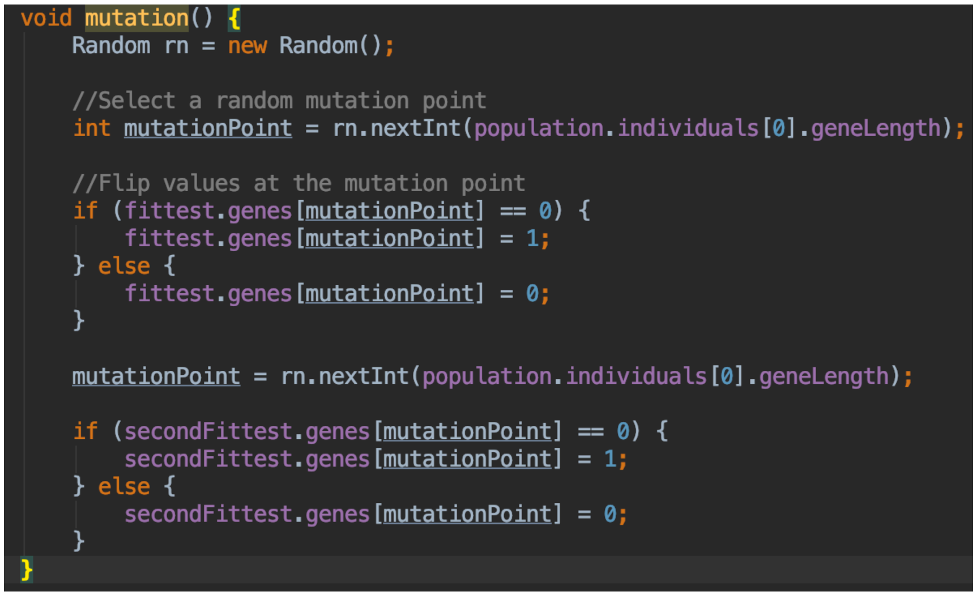
```
//Selection
void selection() {

    //Select the most fittest individual
    fittest = population.getFittest();

    //Select the second most fittest individual
    secondFittest = population.getSecondFittest();
```

4) Crossover:



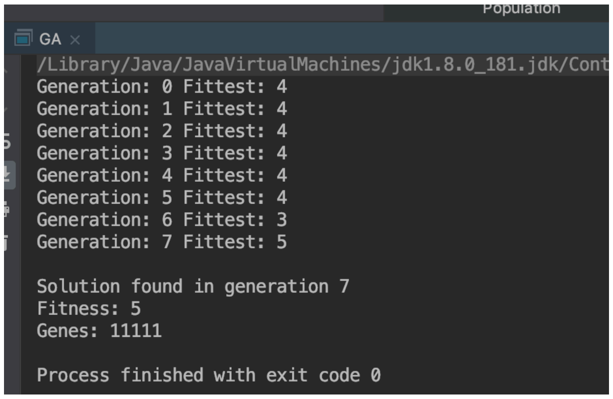
```
//Crossover
void crossover() {
    Random rn = new Random();

    //Select a random crossover point
    int crossOverPoint = rn.nextInt(population.individuals[0].geneLength);

    //Swap values among parents
    for (int i = 0; i < crossOverPoint; i++) {
        int temp = fittest.genes[i];
        fittest.genes[i] = secondFittest.genes[i];
        secondFittest.genes[i] = temp;
```

5) Mutation:



```
void mutation() {
    Random rn = new Random();

    //Select a random mutation point
    int mutationPoint = rn.nextInt(population.individuals[0].geneLength);

    //Flip values at the mutation point
    if (fittest.genes[mutationPoint] == 0) {
        fittest.genes[mutationPoint] = 1;
    } else {
        fittest.genes[mutationPoint] = 0;
    }

    mutationPoint = rn.nextInt(population.individuals[0].geneLength);

    if (secondFittest.genes[mutationPoint] == 0) {
        secondFittest.genes[mutationPoint] = 1;
    } else {
        secondFittest.genes[mutationPoint] = 0;
    }
}
```

c) Compile and run the code, explain the result.



```
/Library/Java/JavaVirtualMachines/jdk1.8.0_181.jdk/Cont
Generation: 0 Fittest: 4
Generation: 1 Fittest: 4
Generation: 2 Fittest: 4
Generation: 3 Fittest: 4
Generation: 4 Fittest: 4
Generation: 5 Fittest: 4
Generation: 6 Fittest: 3
Generation: 7 Fittest: 5

Solution found in generation 7
Fitness: 5
Genes: 11111

Process finished with exit code 0
```

In this question, it will generate the next generation, and check their fittest. For each one, its Fittest is 6-digit number, and count the "1", if the number over 5, that can be stop.

4. Read this paper "Genetic Algorithms for Balanced Minimum Spanning Tree Problem".

Note: Read and understand only the first 5 pages.

https://annals-csis.org/Volume_5/pliks/249.pdf

5. Read the following links on Genomics. Nothing to report except it would be on next quiz.

https://ghr.nlm.nih.gov/primer/mutationsanddisorders/possiblemutations
https://ghr.nlm.nih.gov/primer/basics/gene
https://www2.le.ac.uk/projects/vgec/highereducation/topics/dna-genes-chromosomes
https://ghr.nlm.nih.gov/primer/mutationsanddisorders/possiblemutations