MID TERM PRESENTATION

TEAM TA

6.14.2023

AGENDA

- Team Members and Development Environment.
- Development Progress.
  - Tasks Completed.
- Dijkstra’s Algorithm.
- Hard Karp Algorithm.
- Class Diagram for Task 3.
- Class Diagram for Task 4.
- Overview of the progress.
- Demonstration and Test Data.
- Contribution.
- Future Plans.

TEAM MEMBERS – TEAM TA

- **Members:**
  Tithira Withanaarachchi

DEVELOPMENT ENVIRONMENT

- **Operating System:** Windows
- **Programming Language:** Java
- **IDE:** VS Code
- **Version Control:** Git Hub

```
ion:absolute;z-index:99
x 5px #ccc}.gbrtl .gbm
;display:block;positio
pacity:1;*top:-2px;*l
/;top:-4px\0/;left:-6
ne-box;display:inline
isplay:block;list-s
ne-block;line-height
pointer;display:bl
tive;z-index:1000}
padding-right:9px
d:url(//
```

DEVELOPMENT PROGRESS

LET'S DIVE IN

THE TASKS COMPLETED

Phase 1

Task 1: Product Information: Hash Maps and Java stream were used to process information.

Task 2: Relevant Pairs of Products: LinkedHashMap, HashMap, and Java Stream are used.

Task 3: Route Simulation: Dijkstra’s Algorithm is used to find the shortest Distance between 2 points.

Phase 2

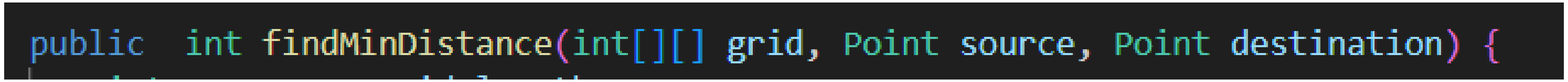
Task 4: Relevant Pairs of products with a large amount of generated test data: Java Random is used to generate test data.

Task 5: Route Simulation(generalized): Dijkstra’s Algorithm and Held Karp Algorithm are used.

DIJKSTRA’S ALGORITHM

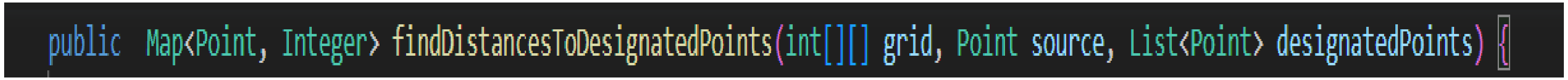
- Main Purpose is to find the shortest distance between two points of a graph.
- Dijkstra’s Algorithm is used in Task 3, Task 5, and Task 6.

Task 3

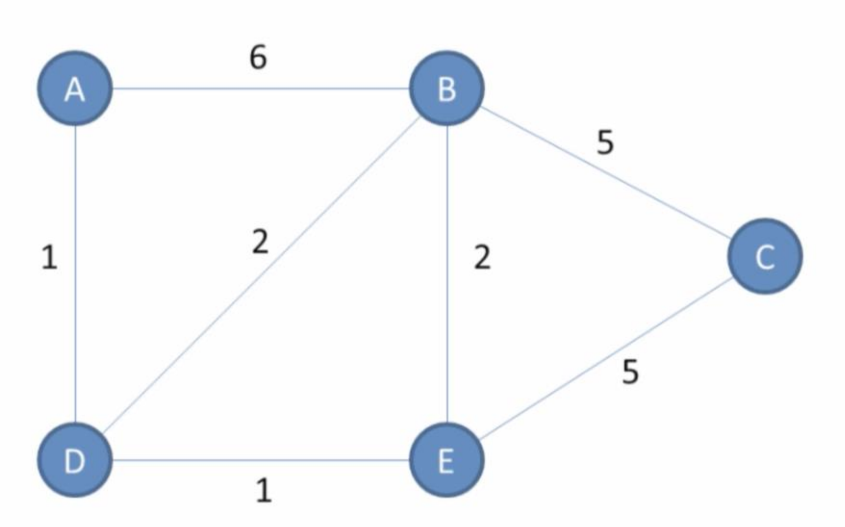


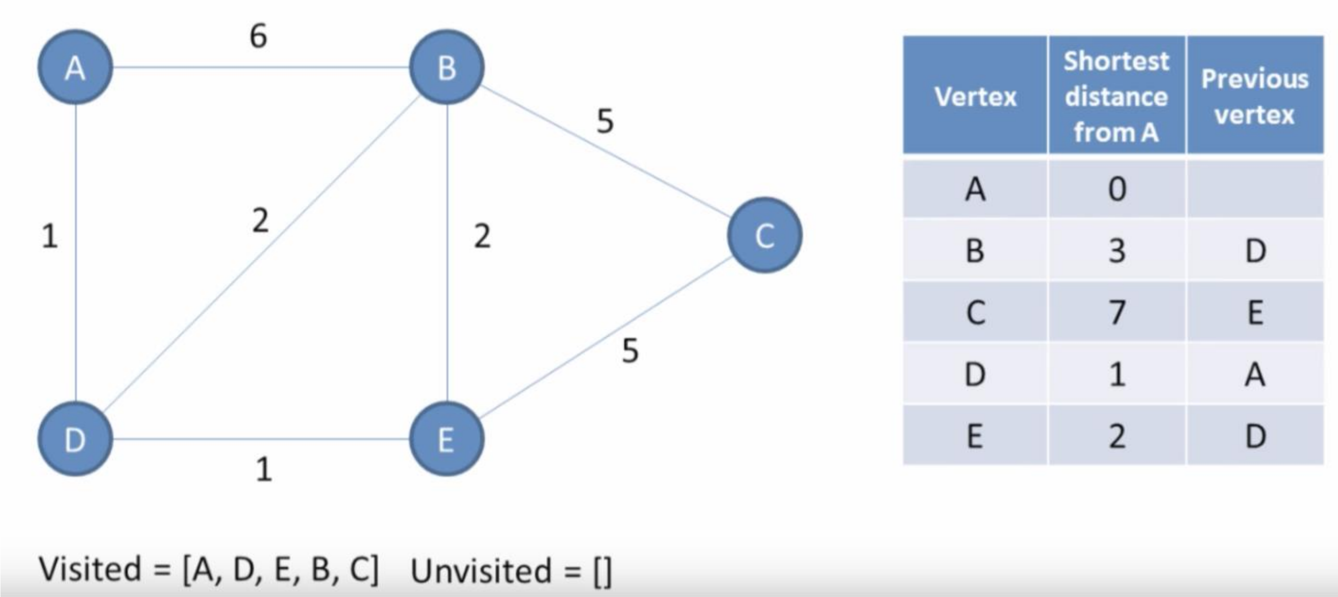
```
public  int findMinDistance(int[][] grid, Point source, Point destination) {
```

Task 5



```
public  Map<Point, Integer> findDistancesToDesignatedPoints(int[][] grid, Point source, List<Point> designatedPoints) {
```





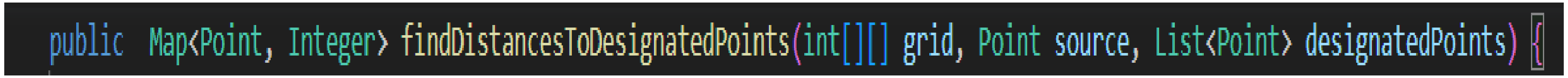
Vertex	Shortest distance from A	Previous vertex
A	0	
B	3	D
C	7	E
D	1	A
E	2	D

Visited = [A, D, E, B, C]   Unvisited = []

HELD KARP ALGORITHM

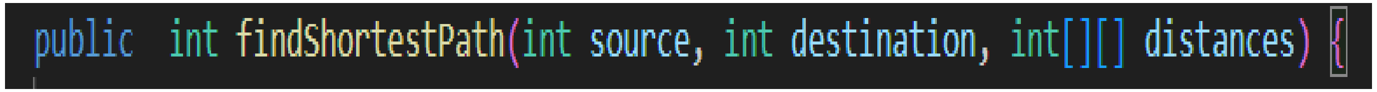
- Held Karp Algorithm is used in Task 5 and Task 6.
- It is used to find the shortest path in order to reach the destination from the source while visiting designated points.

Dijkstra's Algorithm



```
public  Map<Point, Integer> findDistancesToDesignatedPoints(int[][] grid, Point source, List<Point> designatedPoints) {
```

Held Karp Algorithm



```
public  int findShortestPath(int source, int destination, int[][] distances) {
```

$$\begin{array}{ccccc} & 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 15 & 6 \\ 1 & 1 & 0 & 7 & 3 \\ 2 & 15 & 7 & 0 & 12 \\ 3 & 6 & 3 & 12 & 0 \end{array}$$

Matrix

{}
{1} {2} {3}
{1,2} {2,3} {1,3}
{1,2,3}

Subsets

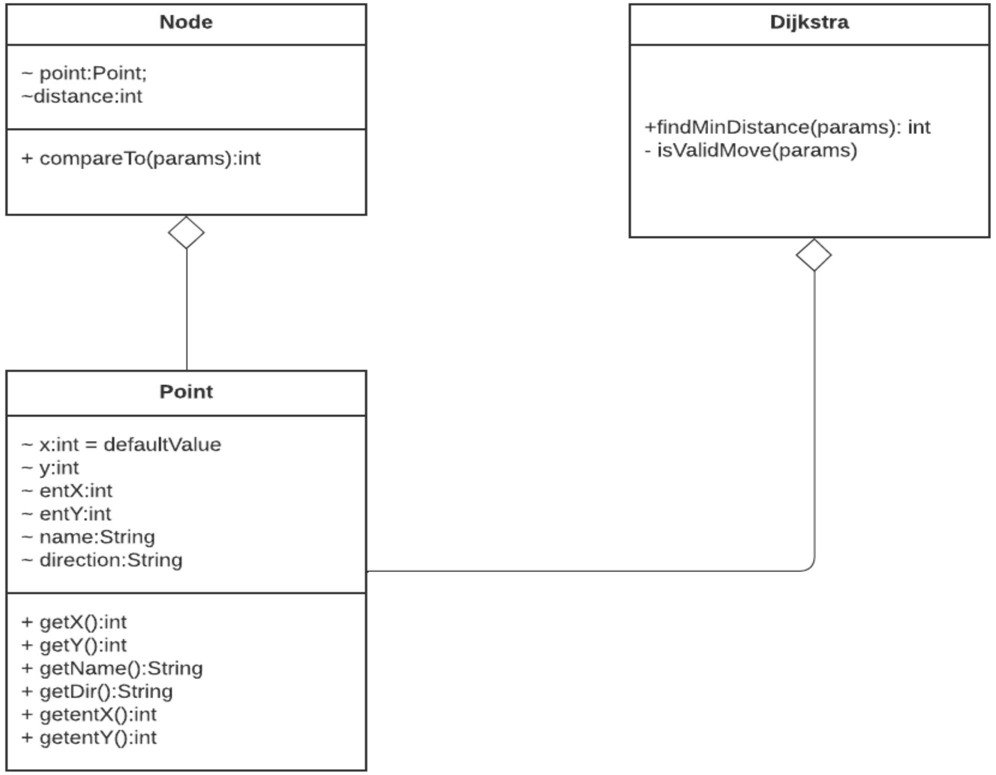
{1,{}}
{2,{}}
{3,{}}
{2,{1}}
{3,{1}}
{1,{2}}
{3,{2}}
{1,{3}}
{2,{3}}

Combinations

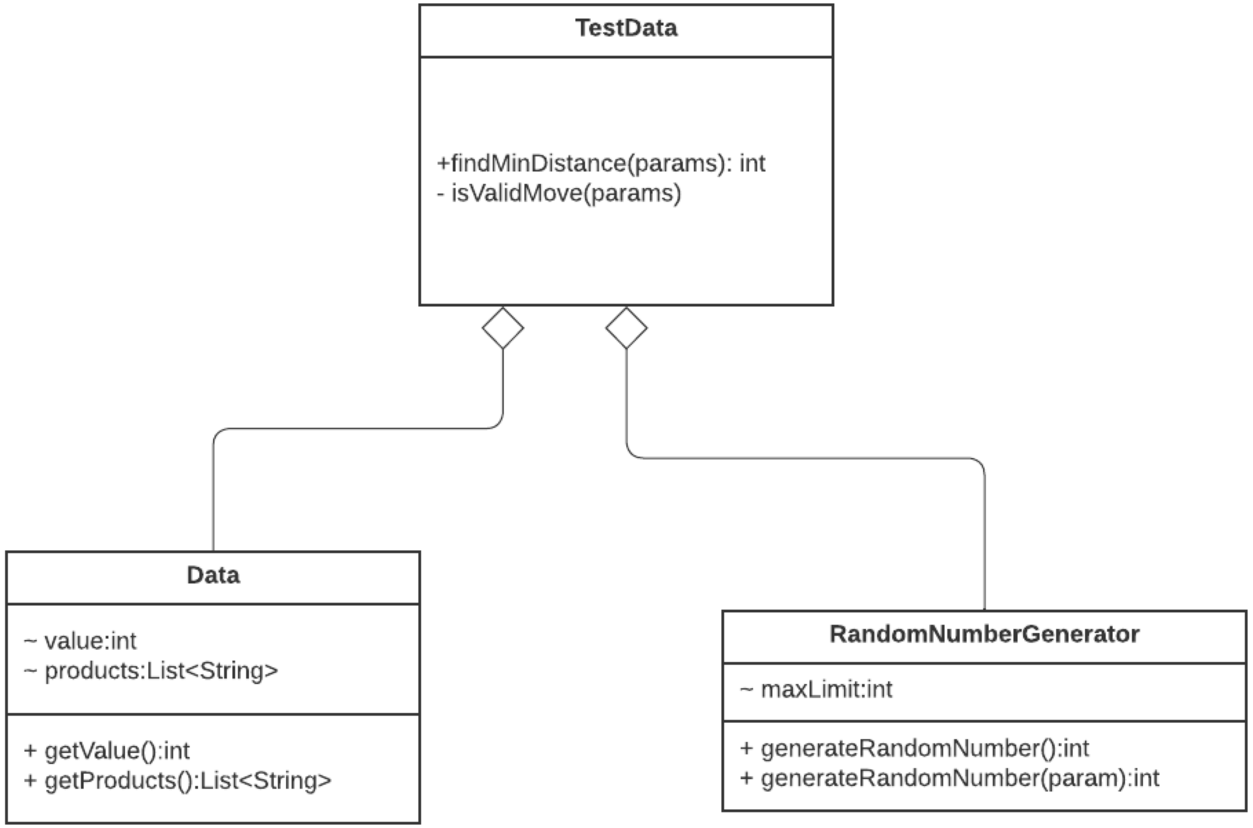
{1,{2,3}}
{2,{1,3}}
{3,{1,2}}

Combinations

CLASS DIAGRAM FOR TASK 3



CLASS DIAGRAM FOR TASK 4


TestData
+findMinDistance(params): int
- isValidMove(params)
Data
~ value:int
~ products:List<String>
+ getValue():int
+ getProducts():List<String>
RandomNumberGenerator
~ maxLimit:int
+ generateRandomNumber():int
+ generateRandomNumber(param):int

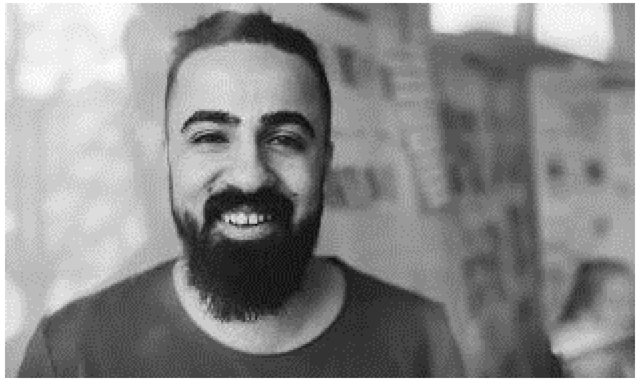
DEMONSTRATION AND TEST DATA

LET’S EXECUTE THE CODE

OVERVIEW OF PROGRESS

Phase 1	
Task 1: Completed	100%
Task 2: Completed	100%
Task 3: Completed	100%
Phase 2	
Task 4: Completed	100%
Task 5: Completed	100%
Task 6: Working On	60%

CONTRIBUTION



Tithira Withanaarachchi: 100%

FUTURE PLANS

- Use a Test Code Visualizer for Demonstration Purposes.

- Complete the Rest of the Tasks that will be assigned, Before the 20th of July.

THANK YOU

Team TA