Task 1. Graph processing.

There is an array of positive integers. The array has to be with equal dimensions, i.e i=j.
The cells in the array are the vertices of the graph. The integers are the edges of the graph.

The first vertex 'initialVertex' is associated with the edge values [0][0], [1][0],.....[array.length-1][0]. You can move through the array only from left to right, from left to diagonal up and from left to diagonal down.

You need to find the longest path from the original 'initialVertex', which is represented in the array. In this case, you need to print all the longest found paths from 'initialVertex'.

Example, given an array:

```
{1, 3, 3},
{2, 1, 4},
{0, 6, 4}
```

output:

```
all longest paths tree from initialVertex:
initialVertex -> [0][0] = 1
initialVertex -> [1][1] = 3
initialVertex -> [2][2] = 12
initialVertex -> [0][1] = 5
initialVertex -> [1][2] = 12
initialVertex -> [0][2] = 8
initialVertex -> [2][0] = 0
initialVertex -> [1][0] = 2
initialVertex -> [2][1] = 8
longest path tree from vertex 'initialVertex' = 12
```

hint: you need to implement Dijkstra's algorithm "in reverse". You can use google guava library.

PS.
This is how we are going to run and evaluate your submission, so please make sure to run below steps before submitting your answer.

- Ensure the application is compiling and runnable, all dependencies are included.
- Commit everything using: git add --all && git commit -m "Task1"
- Create patch file with name without spaces 'GP-java_Task1_<YourNameHere>.patch', using the specified tag as the starting point:

git format-patch initial-commit --stdout > GP-java_Task1_<YourNameHere>.patch)

- Store your file in a shared location where the GP team can access and download it for evaluation. Submit your sharable link to the GP team.