Program znajdujący wyjście z labiryntu

Autorem programu jest Krzysztof Nowakowski. Projekt wykonywany był w grupie razem z Damianem Guzikiem. Moja część programu odpowiada za znajdowania ścieżki wyjścia z labiryntu.

Do znajdowania wyjścia z labiryntu użyty został algorytm Przeszukiwania wszerz (*ang. breadth-first search, BFS*) jest to jeden z najprostszych algorytmów przeszukiwania grafu. Przechodzenie grafu rozpoczyna się od zadanego wierzchołka s i polega na odwiedzeniu wszystkich osiągalnych z niego wierzchołków. Wynikiem działania algorytmu jest drzewo przeszukiwania wszerz o korzeniu w s, zawierające wszystkie wierzchołki osiągalne z s. Do każdego z tych wierzchołków prowadzi dokładnie jedna ścieżka z s, która jest jednocześnie najkrótszą ścieżką w grafie wejściowym.

Wybór tego algorytmu jest przede wszystkim uzasadniony łatwością w implementacji i wystarczającą, jak dla tego projetku, złożonością:

* Czasową: ,
* Pamięciową: ,

Gdzie V to ilość wierzchołków w grafie, a E to ilość krawędzi w grafie.

Do implementacji algorytmu została użyta kolejka FIFO (*ang. First In, First Out*), w programie jest ona reprezentowana przy pomocy *PriorityQueue* pochodzącej z biblioteki *java.util.* Została ona użyta ze wzglądu na szybkość dodawania i usuwania kolejnych elementów z kolejki, ponieważ są to operacje dominujące w tym algorytmie.

Danymi wejściowymi dla aprogramu jest graf w postaci listy sąsiedztwa wierzchołków, z którego pobierany jest wierzchołek od którego rozpoczynane jest przeszukiwanie, jest on jednosześnie wejściem do labiryntu.

Danymi wyjściowymi jest najkrótsza istniejąca ścieżka pomiędzy wejściem a wyjściem z labiryntu.

Uruchomienie:

Należy skompilowac pliki w folderze maze a następnie uruchomić mainProgram. Po otwarciu programu w konsoli podajemy wymagane dane tj. Rozmiar labiryntu i punkt startowy a następnie obserwujemy wyniki w konsoli i oknie. W konsoli wypisana zostanie lista sąsiedztwa każdego wierzchołka a w oknie będzie można zaobserować wygenerowany algorytm i krok po kroku działanie algorytmu szukającego wyjścia.