adjacencyList - списки смежности - запись ребер графа в таблице (двумерный массив) вида:

|  |  |  |
| --- | --- | --- |
| Вершина 1 | Вершина 2 | Стоимость ребра |
| 0 | 1 | 7 |
| … | … | … |

//ребро АВ со стоимостью 7

В программе если есть ребро АВ, то мы записываем только это ребро. Т.е. мы не вносим ребро ВА, которое точно такое же, только направлено в другую сторону, потому что все ребра у нас направлены в обе стороны и в обе стороны стоимость прохождения одинакова.

nodesNum - количество вершин

example graph – пример графа из википедии для алгоритма Прима

spanningTree – остовное дерево

priceHeap – куча, в которой хранятся стоимость ветвей графа, позволяет быстрее найти min или max, это лишнее – потом уберу

Всё что отмечено комментарием temporary – потом будет удалено или доработано

priorityQueueMap

Map – дословно словарь, пара (ключ, значение)

В этом списке мы храним приоритетную очередь (priority queue) ребер на добавление в новый граф в алгоритме

PrimsAlgorithm – алгоритм, который строит остовное дерево

========================================================================================

Задача:

Построить исходный граф adjacencyList в конструкторе graph() класса graph

========================================================================================