**КФУ, Институт ИТИС, 1 курс, 2 семестр, 2021**

**Семестровая работа №2:**

**«Алгоритм Прима.**

**Построение минимального остовного дерева»**

**Выполнила:**

Рыфтина Полина Алексеевна

группа 11-103

**Преподаватель:**

Андреичев М.Д.

*Алгоритм Прима*

**Алгоритм Прима** — алгоритм построения минимального остовного дерева взвешенного связного неориентированного графа.

*Историческая справка*

Алгоритм впервые был открыт в 1930 году чешским математиком Войцехом Ярником, позже переоткрыт Робертом Примом в 1957 году, и, независимо от них, Э. Дейкстрой в 1959 году.

*Описание*

На вход алгоритма подаётся связный неориентированный граф. Для каждого ребра задаётся его стоимость.

Сначала берётся произвольная вершина и находится ребро, инцидентное данной вершине и обладающее наименьшей стоимостью. Найденное ребро и соединяемые им две вершины образуют дерево. Затем, рассматриваются рёбра графа, один конец которых — уже принадлежащая дереву вершина, а другой — нет; из этих рёбер выбирается ребро наименьшей стоимости. Выбираемое на каждом шаге ребро присоединяется к дереву. Рост дерева происходит до тех пор, пока не будут исчерпаны все вершины исходного графа.

Результатом работы алгоритма является остовное дерево минимальной стоимости.

*Пример*

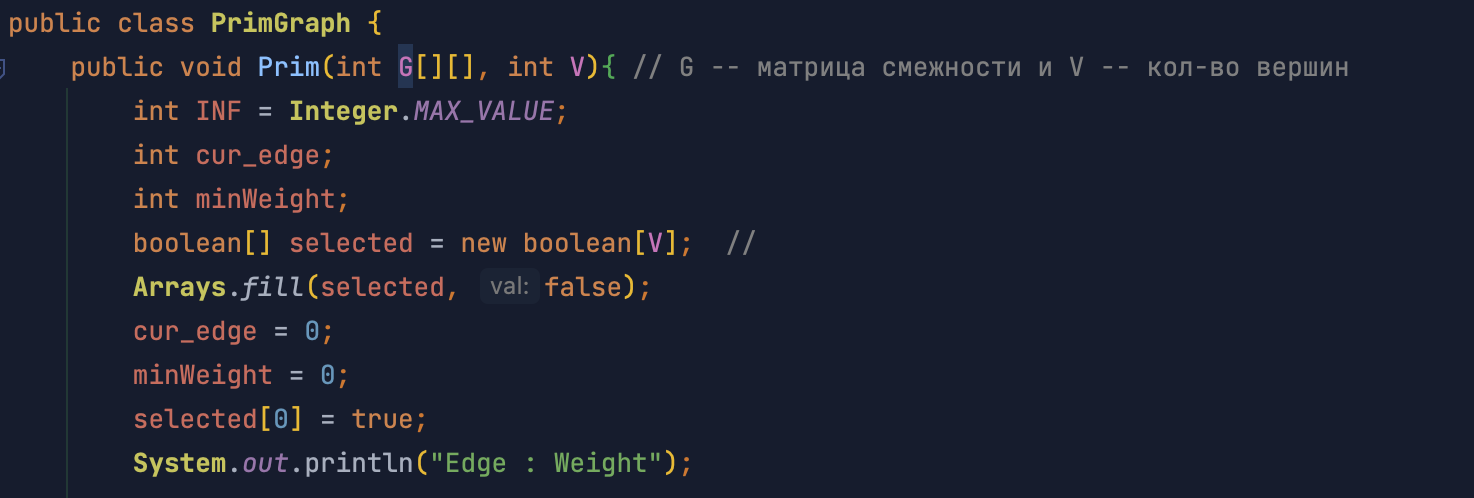
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Изображение** | **Множество выбранных вершин U** | **Ребро**  **(u, v)** | **Множество невыбранных вершин V \ U** | **Описание** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | {} |  | {A,B,C,D,E,F,G} | Исходный взвешенный граф. Числа возле ребер показывают их веса, которые можно рассматривать как расстояния между вершинами. |
|  | {D} | (D,A) = 5 **V**  (D,B) = 9 (D,E) = 15 (D,F) = 6 | {A,B,C,E,F,G} | В качестве начальной произвольно выбирается вершина **D**. Каждая из вершин **A**, **B**, **E** и **F** соединена с **D** единственным ребром. Вершина **A** — ближайшая к **D**, и выбирается как вторая вершина вместе с ребром **AD**. |
|  | {A,D} | (D,B) = 9 (D,E) = 15 (D,F) = 6 **V** (A,B) = 7 | {B,C,E,F,G} | Следующая вершина — ближайшая к любой из выбранных вершин **D** или **A**. **B** удалена от **D** на 9 и от **A** — на 7. Расстояние до **E** равно 15, а до **F** — 6. **F** является ближайшей вершиной, поэтому она включается в дерево **F**вместе с ребром **DF**. |
|  | {A,D,F} | (D,B) = 9 (D,E) = 15 (A,B) = 7 **V** (F,E) = 8 (F,G) = 11 | {B,C,E,G} | Аналогичным образом выбирается вершина **B**, удаленная от **A** на 7. |
|  | {A,B,D,F} | (B,C) = 8 (B,E) = 7 **V** (D,B) = 9 цикл (D,E) = 15 (F,E) = 8 (F,G) = 11 | {C,E,G} | В этом случае есть возможность выбрать либо **C**, либо **E**, либо **G**. **C**удалена от **B** на 8, **E** удалена от **B** на 7, а **G** удалена от **F** на 11. **E** — ближайшая вершина, поэтому выбирается **E** и ребро **BE**. |
|  | {A,B,D,E,F} | (B,C) = 8 (D,B) = 9 цикл (D,E) = 15 цикл (E,C) = 5 **V** (E,G) = 9 (F,E) = 8 цикл (F,G) = 11 | {C,G} | Здесь доступны только вершины **C** и **G**. Расстояние от **E** до **C** равно 5, а до **G** — 9. Выбирается вершина **C** и ребро **EC**. |
|  | {A,B,C,D,E,F} | (B,C) = 8 цикл (D,B) = 9 цикл (D,E) = 15 цикл (E,G) = 9 **V** (F,E) = 8 цикл (F,G) = 11 | {G} | Единственная оставшаяся вершина — **G**. Расстояние от **F** до неё равно 11, от **E** — 9. **E** ближе, поэтому выбирается вершина **G** и ребро **EG**. |
|  | {A,B,C,D,E,F,G} | (B,C) = 8 цикл (D,B) = 9 цикл (D,E) = 15 цикл (F,E) = 8 цикл (F,G) = 11 цикл | {} | Выбраны все вершины, минимальное остовное дерево построено (выделено зелёным). В этом случае его вес равен 39. |

*Реализация*

Основные моменты:

* На вход функция Prim() принимает G – матрицу смежности и V – кол-во вершин;
* Selected – массив для отслеживания помеченных вершин;
* Количество ребер в минимальном связном дереве всегда будет меньше (V -1), где V - количество вершин;
* selected[0] – будет всегда начальной вершиной;
* Для каждой вершины в selected ищутся все смежные вершины и вычисляются расстояния (если расстояние до смежной вершины минимально из всех, то она добавляется в selected)





*Оценка времени*

Асимптотика алгоритма зависит от способа хранения графа и способа хранения помеченных вершин.

|  |  |  |
| --- | --- | --- |
| **Способ представления помеченных вершин и графа** | **Асимптотика** | **Стоимость операции присвоения метки вершине** |
| Массив и двумерный массив | *O(V2)* | *O(1)* |
| Бинарная пирамида и двумерный массив | *O((V + E) +logV) =O(E logV)* | *O(logn)* |
| Фибоначчиева пирамида и двумерный массив | *O(logn)* | *O(1)* |

Замеры производились на 100 матрицах смежности, размером от 1\*1 до 496\*496. Вес ребер генерировался произвольным образом от 0 до 1000. В моей работе представлен первый случай.

Неровности в графиках вызваны случайной генерацией данных (длина строки от 1 до 1000) и нестабильной работой процессора из-за нерегулирующихся сторонних процессов на моем компьютере.