Лабораторная работа №1

Отыскание кратчайшего пути в ориентированном графе между двумя заданными вершинами методом Дейкстры.

Математической моделью структуры вычислительной сети, отображающей ее топологию, является граф. Поэтому задачи теории графов имеют непосредственное отношение к вопросам построения, использования и оценки вычислительных сетей. Одним из приложений теории графов является решение задачи маршрутизации, а именно отыскания наиболее оптимального маршрута для сообщения, посылаемого от одного узла сети к другому.

Исходные данные. Дан граф $G=(X,\Gamma)$, дугам которого приписаны веса, задаваемые матрицей $C=|c_{ij}|$. Веса могут быть интерпретированы по-разному, в зависимости от постановки задачи: как показатели надежности, пропускной способности, стоимости. В данном случае будем интерпретировать веса как расстояния.

Матрица C попутно дает также информацию о смежности вершин и направленности дуг.

Будем считать, что граф не содержит циклы с отрицательными суммарными весами.

Постановка задачи. Задача о кратчайшем пути состоит в нахождении кратчайшего пути от заданной начальной вершины $s \in X$ до заданной конечной вершины $t \in X$, при условии, что такой путь существует, то есть при условии $t \in R(s)$, где $R(s)$ - множество вершин, достижимых из s.

Метод Дейкстра. Наиболее эффективный алгоритм решения задачи о кратчайшем $(s-t)$-пути для неотрицательной матрицы весов $(c_{ij} \geq 0, \quad \forall i, j)$ был дан Дейкстра. В общем случае этот метод основан на приписывании вершинами временных пометок, причем пометка вершины дает верхнюю границу длины пути от s к этой вершине. Эти пометки (их величины) постепенно уменьшаются с помощью некоторой итерационной процедуры, и на каждом шаге итерации точно одна из временных пометок становится постоянной. Последнее указывает на то, что что пометка уже не является верхней границей, а дает точную длину кратчайшего пути от s к рассматриваемой вершине.

Алгоритм Дейкстры.

Пусть $l(x_i)$ - пометка вершины x_i.

Присвоение начальных значений

Шаг 1. Положить $l(s)=0$ и считать эту пометку постоянной.

Положить $l(x_i)=\infty$ для всех $x_i \neq s$ и считать эти пометки временными. Положить $p=s$.

Обновление пометок

Шаг 2. Для всех $x_i \in \Gamma(p)$, пометки которых временные, изменить пометки в соответствии со следующим выражением

$$l(x_i) \leftarrow \mathbf{min}[l(x_i), \quad l(p)+c(p,x_i)]$$

Превращение пометки в постоянную

Шаг 3. Среди всех вершин с временными пометками найти такую, для которой $l(x_i^*)=\mathbf{min}[l(x_i)]$.

Шаг 4. Считать пометку вершины x_i^* постоянной и положить $p=x_i^*$.

Шаг 5. а) *(выполняется, если надо найти лишь путь от* s *к* t*)* Если $p=t$, то $l(p)$ является длиной кратчайшего пути. Останов. Если $p \neq t$, перейти к шагу 2.

б) *(выполняется, если требуется найти пути от* s *ко всем остальным вершинам*) Если все вершины помечены как постоянные, то эти пометки дают длины кратчайших путей. Останов. Если некоторые пометки являются временными, перейти к шагу 2.

Как только длины кратчайших путей от будут найдены, сами пути можно получить при помощи рекурсивной процедуры с использованием соттношения (*). Так как вершина $x_i^{'}$ непосредственно предшествует вершине x_i в кратчайшем пути от s к x_i, то для любой вершины x_i соответствующую вершину $x_i^{'}$ можно найти как одну из оставшихся вершин, для которой

$$l(x_i^{'})+c(x_i^{'},x_i)=l(x_i)\text{. (*)}$$

Если кратчайший путь от s до любой вершины x_i является единственным, то дуги этого кратчайшего пути ($x_i^{'},x_i$) образуют ориентированное дерево с корнем s.

Если кратчайших путей несколько, то выбор конкретного пути может быть произвольным. В этом случае кратчайшие пути образуют граф, называемый базой относительно s, или s-базой.

Пример. Рассмотрим алгоритм на примере графа, представленного на крисунке. Неориентированное ребро представляет собой замену двух противоположных дуг равного веса. Требуется найти все кратчайшие пути от вершины x_1 ко всем остальным. Матрица весов приведена ниже.

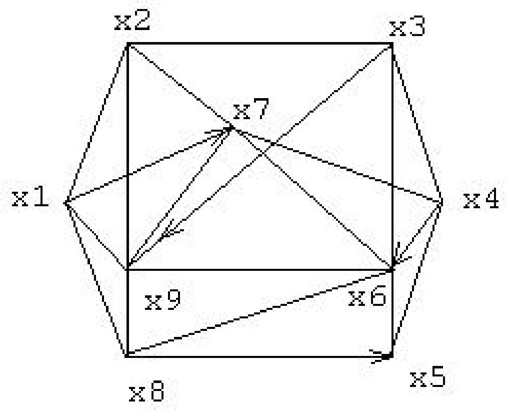

Рисунок 1

$$
\begin{array}{cc}
 & \begin{array}{ccccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 \end{array} \\
C = \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{array} &
\left|\begin{array}{ccccccccc}
 & 10 & & & & & 3 & 6 & 12 \\
10 & & 18 & & & & 2 & & 13 \\
 & 18 & & 25 & & 20 & & & 7 \\
 & & 25 & & 5 & 16 & 4 & & \\
 & & & 5 & & 10 & & & \\
 & & 20 & & 10 & & 14 & 15 & 9 \\
 & 2 & & 4 & & 14 & & & 24 \\
6 & & & & 23 & 15 & & & 5 \\
12 & 13 & & & & 9 & 24 & 5 &
\end{array}\right|
\end{array}
$$

В процессе выполнения алгоритма Дейкстры постоянные пометки будут снабжаться знаком +, остальные пометки рассматриваются как временные.

Первые три итерации работы алгортма:

Шаг 1. $l(x_1) = 0^+, \quad l(x_i) = \infty \quad \forall x_i \neq x_1, \quad p = x_1$

Первая итерация

Шаг 2. $\Gamma(p) = \Gamma(x_1) = \{x_2, x_7, x_8, x_9\}$, все пометки временные. Для x_2 получим:

$$l(x_2) = \mathbf{min}[\infty, 0^+ + 10] = 10,$$

аналогично $l(x_7) = 3, \quad l(x_8) = 6, \quad l(x_9) = 12$.

Шаг 3. Ищем минимальную временную пометку: $\mathbf{min}(10, \infty, \infty, \infty, \infty, 3, 6, 12) = 3$ соответствует x_7.

Шаг 4. x_7 получает постоянную пометку $l(x_7) = 3^+$, $p = x_7$.

Шаг 5. Не все вершины имеют постоянные пометки, поэтому переходим к шагу 2.

Вектор пометок: $(0^+, 10, \infty, \infty, \infty, \infty, 3^+, 6, 12)$.

Вторая итерация

Шаг 2. $\Gamma(p) = \Gamma(x_7) = \{x_2, x_4, x_6, x_9\}$, все пометки временные. Для x_2 получим:

$$l(x_2) = \mathbf{min}[10, 3^+ + 2] = 5,$$

аналогично $l(x_4) = 7, \quad l(x_6) = 17, \quad l(x_9) = 12$.

Шаг 3. $\mathbf{min}(5, \infty, 7, \infty, 17, 6, 12) = 5$ соответствует x_2.

Шаг 4. x_2 получает постоянную пометку $l(x_2) = 5^+$, $p = x_2$.

Шаг 5. Не все вершины имеют постоянные пометки, поэтому переходим к шагу 2.

Вектор пометок: $(0^+, 5^+, \infty, 7, \infty, 17, 3^+, 6, 12)$.

Третья итерация

Шаг 2. $\Gamma(p) = \Gamma(x_2) = \{x_1, x_3, x_7, x_9\}$, только вершины x_3 и x_9 имеют временные пометки. Для x_3 получим:

$$l(x_3) = \mathbf{min}[\infty, 5^+ + 18] = 23,$$

аналогично $l(x_9) = 12$.

Шаг 3. $\mathbf{min}(23, 7, \infty, 17, 6, 12) = 6$ соответствует x_8.

Шаг 4. x_8 получает постоянную пометку $l(x_8) = 6^+$, $p = x_8$.

Шаг 5. Не все вершины имеют постоянные пометки, поэтому переходим к шагу 2.

Вектор пометок: $(0^+, 5^+, 23, 7, \infty, 17, 3^+, 6^+, 12)$.

Продолжая этот процесс, получим окончательную разметку $(0^+, 5^+, 18^+, 7^+, 12^+, 17^+, 3^+, 6^+, 11^+)$.

Задание

1. Ознакомиться с алгоритмом Дейкстры.
2. Разработать программу, которая позволяет решить задачу отыскания кратчайшего пути между любой парой вершин

ориентированного графа с помощью алгоритма Дейкстры.

Требования к программе:

- количество вершин графа вводится пользователем и заранее неизвестно (не более 10);

- граф задается матрицей иницидентности, которая также выполняет роль матрицы весов; в графе не должно быть циклов с отрицательным весом;

- вершины, между которыми ищется кратчайший путь, задаются пользователем и заранее неизвестны;

- для заданной пары вершин программа должна либо отыскать кратчайший путь между ними, и при этом выдать:

а) длину пути;

б) список вершин, через которые проходит кратчайший путь (необходимо также выделить путь на графе);

либо выдать сообщение, что пути между вершинами не существует.

3. Требования к интерфейсу программы. Необходимо предусмотреть две возможности задания графа и внесения корректировок в него:

- с помощью матрицы инцидентности, которая доступна пользователю для заполнения и редактирования;

- непосредственно с помощью набора операций над вершинами и дугами графа, которые вызываются по событиям, связанным с щелчками мышью или нажатием клавиш.

Например, нажатие правой клавиши мыши в той части экрана, в которой выведен граф, приводит к появлению контекстного меню, пункты которого позволяют:

- добавить вершину в граф;
- добавить дугу;
- удалить вершину;
- удалить дугу;
- изменить вес дуги.

Требуется также предусмотреть возможность перетаскивания вершин графа, а также задания исходной и конечной вершин путем их выбора с помощью мыши.