**Графы. Представление и обходы**

Граф – это множество однотипных **объектов** (вершин), некоторые из которых **связаны** друг с другом какими-либо связями (ребрами). Одна связь всегда соединяет только две вершины (иногда – вершину саму с собой).

Основные разновидности графов:

* **неориентированные** (обычные), в которых важен только **сам факт** связи двух вершин (рис.1.а)
* **ориентированные** (орграфы), для которых важным является еще и **направление** связи вершин (рис.1.б)
* **взвешенные**, в которых важной информацией является еще и **степень** (величина, вес) связи вершин (рис.1.с)

|  |  |  |
| --- | --- | --- |
|  |  |  |
| а) обычный | б) ориентированный | в) взвешенный |

Рисунок 1 - Примеры графов разных типов

**Основные способы представления графов**

а) графическое изображение;

б) указанием множества вершин и множества ребер (дуг);

в) матрица смежности;

г) матрица инцидентности;

д) иерархический список.

**Матрицы смежности**

Матрицы смежности и инцидентности удобны для задания графов средствами ВТ.

***Матрица смежности*** Sm - это квадратная матрица размером N x N (N - количество *вершин* в *графе*), заполненная по следующему правилу:

**Если в** *графе* **имеется** *ребро* e, **соединяющее** *вершины* u **и** v,   
**то** Sm[u,v] = Ves(e), **в противном случае** Sm[u,v] = “-”.

Для рассмотренных выше примеров (рис.1) матрицы смежности будут следующими:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | A | B | C | D | E | | A | 0 | 1 | 1 | 1 | 1 | | B | 1 | 0 | 0 | 0 | 1 | | C | 1 | 0 | 0 | 1 | 0 | | D | 1 | 0 | 1 | 0 | 0 | | E | 1 | 1 | 0 | 0 | 0 | | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | A | B | C | D | E | | A | 0 | 1 | 1 | 0 | 0 | | B | 1 | 0 | 0 | 0 | 1 | | C | 0 | 1 | 0 | 1 | 1 | | D | 1 | 0 | 0 | 0 | 1 | | E | 0 | 0 | 0 | 0 | 0 | | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | A | B | C | D | E | | A | 0 | 15 | 0 | 9 | 0 | | B | 15 | 0 | 10 | 0 | 0 | | C | 0 | 10 | 0 | 22 | 6 | | D | 9 | 0 | 22 | 0 | 19 | | E | 0 | 0 | 6 | 19 | 0 | |
| а) обычный | б) ориентированный | в) взвешенный |

**Основные свойства матриц смежности и инцидентности**

* Матрица смежности неориентированного графа является симметричной. Для ориентированного графа это, вообще говоря, неверно.
* Сумма элементов *i* – ой строки или *i* –го столбца матрицы смежности неориентированного графа равна степени вершины *xi*.
* Сумма элементов *i* – ой строки матрицы смежности ориентированного графа равна числу дуг, исходящих из *xi*.
* Сумма элементов *i* – го столбца матрицы смежности ориентированного графа равна числу дуг, входящих в вершину *xi*.
* Сумма строк матрицы инцидентности ориентированного графа является нулевой строкой.

**Недостатки:**

* заранее надо знать хотя бы ориентировочное число вершин в графе;
* для графов с большим числом вершин матрица становится слишком большой (например, 1000\*1000 = 1 миллион элементов);
* при малом числе связующих ребер матрица заполнена в основном нулями (разреженная матрица).

**Иерархический список**

Недостатков представления графов в виде матриц лишен способ, основанный на использовании иерархических списков. Первый список представляет собой **основной список всех вершин;** множество **вспомогательных списков**, которые содержат перечень смежных вершин или указатели на эти вершины.

**Вершина** = запись (Структура)

Номер: Число;

Имя: Строка;

След Вершина: указатель на **Вершина**;

Список Дуг: **Дуга**;

end;

**Дуга** = запись (Структура)

Вес: Число;

Конец Дуги: **Вершина**;

След Дуга: **Дуга**;

end;

Преимущества иерархического списка

Гибкость и экономичное использовании памяти. И даже небольшая избыточность данных, которая появляется в случае неориентированного графа, задавая каждое ребро как две дуги, компенсируется гибкостью всей структуры, что особенно удобно при необходимости частых перестроений в процессе работы программы.

Для рассмотренных выше обычного графа и ориентированного графа (рис.1 а) 1б) иерархические списки будут следующими:

|  |  |
| --- | --- |
|  |  |

При обработке графов часто приходится выполнять **обход** всех его вершин. Существуют два основных правила обхода, известные как поиск в глубину и поиск в ширину.

**Обход в глубину**

Поиск в глубину использует две структуры – **стек** для запоминания еще не обработанных вершин и **список** для запоминания уже обработанных. Поиск выполняется следующим образом:

1. задать стартовую вершину (аналог корневой вершины при обходе дерева)
2. обработать стартовую вершину и включить ее во вспомогательный список обработанных вершин
3. включить в стек все вершины, смежные со стартовой
4. организовать цикл по условию *опустошения* стека и внутри цикла выполнить:
   * извлечь из стека очередную вершину
   * проверить по вспомогательному списку обработанность этой вершины
   * если вершина уже обработана, то извлечь из стека следующую вершину
   * если вершина еще не обработана, то обработать ее и поместить в список обработанных вершин
   * просмотреть весь список смежных с нею вершин и поместить в стек все еще не обработанные вершины

Пример 1. Организация обхода графа рис.1.а; стартовая вершина – B.

| **№** | **Список** | **Стек** | **Пояснение** |
| --- | --- | --- | --- |
|  | (В) | (А, Е)  → (head) | * + 1. включаем B в список обработанных вершин; помещаем в стек смежные с В вершины, т.е. A и E. |
|  | (В, Е) | (А) | * + 1. извлекаем из стека вершину E; т.к. E нет в списке обработанных вершин, то обрабатываем ее и помещаем в список. |
|  | (В, Е) | (А, А) | * + 1. смежные с E вершины – это A и B, но B уже обработана, поэтому помещаем в стек только вершину А. |
|  | (В, Е, А) | (А) | * + 1. извлекаем из стека вершину А, т.к. А нет в списке обработанных вершин, то помещаем ее туда |
|  | (В, Е, А) | (A, C, D) | смежные с А вершины – это B, C, D, E, из которых B и E уже обработаны, поэтому помещаем в стек C и D. |
|  | (B, E, A, D) | (A, C) | извлекаем из стека вершину D, т.к. D не обработана, то помещаем ее в список. |
|  | B, E, A, D) | (А, С, С) | смежные с D вершины – это А и С, из которых А уже обработана, поэтому помещаем в стек вершину С. |
|  | (B, E, A, D, C) | (А, С) | извлекаем из стека вершину С, и т.к. С не обработана, помещаем ее в список. |
|  | (B, E, A, D, C) | (A, С) | смежные с С вершины – это A и D, но они обе уже обработаны, поэтому в стек ничего не заносим. |
|  | (B, E, A, D, C) | ( A ) | извлекаем из стека С, но она уже обработана, в список ничего не помещаем. |
|  | (B, E, A, D, C) | () | извлекаем из стека А, но она тоже уже обработана, стек стал пустым, то завершаем обход с результатом (B, E, A, D, C) |

**Поиск в ширину**

Организация обхода отличается тем, что сначала обрабатываются все вершины, **смежные** с текущей, а лишь потом –**потомки вершин**. Вместо стека для хранения еще не обработанных вершин используется **очередь**.

Последовательность действий:

* + 1. задать стартовую вершину
    2. обработать стартовую вершину и включить ее во вспомогательный список обработанных вершин
    3. включить в очередь все вершины, смежные со стартовой
    4. организовать цикл по условию опустошения очереди и внутри цикла выполнить:
  + извлечь из очереди очередную вершину
  + проверить по вспомогательному списку обработанность этой вершины
  + если вершина уже обработана, то извлечь из очереди следующую вершину
  + если вершина еще не обработана, то обработать ее и поместить в список обработанных вершин
  + просмотреть весь список смежных с нею вершин и поместить в очередь все еще не обработанные вершины

Пример 2. Организация обхода графа рис.1.а; стартовая вершина – B.

| **№** | **Список** | **Очередь** | **Пояснение** |
| --- | --- | --- | --- |
|  | (В) | (А, Е)  → (head) | включаем B в список обработанных вершин; помещаем в очередь смежные с В вершины, т.е. A и E. |
|  | (В, А) | (Е) | извлекаем из очереди вершину А, т.к. она не обработана, добавляем ее в список. |
|  | (В, А) | (E, C, D, E) | смежные с А вершины – это B, C, D и E, помещаем в очередь вершины C, D и E. |
|  | (B, A, E) | (C, D, E) | извлекаем из очереди вершину Е, т.к. Е не обработана, помещаем ее в список.  ! *в первую очередь обработаны обе смежные с В вершины!* |
|  | (B, A, E) | (C, D, E) | смежные с Е вершины – это А и В, но обе они уже обработаны, поэтому очередь новыми вершинами не пополняется. |
|  | (B, A, E, С) | (D, E) | извлекаем из очереди вершину С, т.к. С не обработана, то помещаем ее в список. |
|  | (B, A, E, С) | (D, E, D) | смежные с С вершины – это А и D, помещаем в очередь только D. |
|  | (B, A, E, С, D) | (E, D) | извлекаем из очереди вершину D, и т.к. D не обработана, помещаем ее в список. |
|  | (B, A, E, С, D) | (E, D) | смежные с D вершины – это А и С, но обе они обработаны, поэтому очередь не пополняется. |
|  | (B, A, E, С, D) | (D) | извлекаем из очереди вершину Е, но она уже обработана: |
|  | (B, A, E, С, D) | () | извлекаем из очереди вершину D, но она уже обработана и т.к. очередь становится пустой, обход заканчивается с результатом (B, A, E, С, D). |

Задание:

1. Построить матрицу смежности для ориентированного графа; примеры графов - “Задания на работу с графами.docx”, ч.1, стр.2-5.
2. Построить матрицу смежности для взвешенного графа; примеры графов - “Задания на работу с графами.docx”, ч.2, стр.6-9. Для данного графа выполнить обходы в глубину и в ширину.