Практикум по курсу теория баз данных (Э.Э.Гасанов)

1. Простейшие структуры данных (разминочные задачи)

1.1. Краткое описание

Этот раздел предназначен для общего понимания структур данных .

Студентам предлагается решить 2 задачи:

1. Реализация одной из простейших структур данных, таких как список, стек, очередь.
2. Реализация сортирующего дерева.

Все входные данные должны считываться из файла и результат выполнения программы выводится в файл.

Написание программы подразумевает включение основных действий :

- добавление элемента,
- удаление элемента,
- редактирование элемента структуры.

Остальные операции с указанной структурой задаются в зависимости от поставленной задачи.

Так же программа должна оценивать время выполнения основных операций.

1.2. Вид решения

Решение должно быть представлено в виде программы (т.е. компилирующегося кода) и отчета о проделанной работе. Который в свою очередь должен содержать файлы выходных данных соответствующие тестовым файлам (содержащим входные данные).

Дополнительное условие заключается в том, что указанная реализация поставленной задачи должна удовлетворять временному интервалу (он указывается в зависимости от конкретной задачи).

Тестовые файлы содержат структурированную последовательность данных (т.е. элементами структуры являются указанные классы, тип которых зависит от поставленной задачи).

Сам тест подразумевает тестовый файл и набор операций со структурой (добавление нового элемента в структуру данных, не требующего проверки на корректность ввода, поиск и удаление элемента структуры, редактирование существующего элемента).

Время работы тестируемой программы фиксируется с момента занесения тестового файла в указанную структуру данных. Оценивается время поиска, редактирования и удаления существующего элемента структуры.

Примеры предлагаемых структур данных содержат списки, стеки, очереди, сортирующие деревья. А также возможны примеры, которые являются различными модификациями изложенных выше структур данных или различные задачи использующие эти структуры.

2. Структуры данных и операции над ними

2.1. Краткое описание

Написать программу, реализующую на основе указанных структур данных ассоциативный массив (типа "целое неотрицательное число в целое неотрицательное число"). Замерить время работы операций с этим массивом для различных реализаций. Операции должны считываться из файла, результаты работы также записываются в файл. Файлы с запросами выдаются. Каждая реализация структуры данных должна реализовывать следующие функции: добавление элемента с заданным ключом и значением, удаление элемента с заданным ключом, поиск элемента с заданным ключом, запись в массив всех значений в порядке, указанном для данной структуры данных (для проверки что реализована именно эта структура данных).

2.2. Вид решения

Решение состоит из:

- программы, т.е. компилируемого исходного кода;
- отчета, сделанного по итогам работы программы.

Программа должна принимать на вход тип реализации (его номер) и файл с запросами. Сначала все запросы должны быть считаны в оперативную память (и преобразованы при этом из текстового формата в бинарный). Затем запросы поочередно обрабатываются. Результаты выполнения запросов записываются в оперативную память. После обработки всех запросов все результаты записываются в файл ответа в текстовом виде. Кроме того, если среди запросов встречаются особые запросы на замеры времени создается файл, содеражащий время работы программы между этими замерами. Также в конце работы программы значения содержащихся в ассоциативном массиве элементов записываются в обычный массив в порядке, указанном для используемой структуры данных. Этот массив записывается в еще один файл. Таким образом в результате обработки файла с запросами программа должна создавать 3 файла (либо 2, если замеров времени не было): файл с ответами, файл с результатами замеров времени и файл с содержанием ассоциативного массива в конце работы программы.

Файлы с запросами являются текстовыми файлами, где каждый запрос записан на отдельной строке. Формат запросов следующий:

1. `insert <ключ> <значение>` – добавление в ассоциативный массив записи с указанными ключом и значением. Если в массиве уже есть запись с таким ключом, то ничего добавлять не требуется.

2. `delete <ключ>` – удалить из ассоциативного массива запись с указанным ключом. Если записи с таким ключом в массиве нет, ничего делать не нужно.

3. `find <ключ>` – найти в ассоциативном массиве запись с указанным ключом.

4. `checkpoint` – произвести замер времени.

Для каждого обработанного запроса (кроме запросов на замер времени) в файле ответа должна содержаться одна строчка с ответом в следующем формате:

1. для запросов на добавление символ `1` если новая запись была добавлена и символ `0` если в массиве уже есть запись с таким ключом;

2. для запросов на удаление – значение удаляемой записи или `-1`, если записи с таким ключом нет;

3. для запросов на поиск – значение для найденной записи или `-1`, если записи с таким ключом нет.

Файл ответа должен заканчиваться пустой строкой.

2.3. Проверка корректности

Для проверки корректности работы программы учащемуся выдается несколько примеров небольших файлов запросов и соответствующих им файлов ответов (+ файлы с правильными конечными состояниями структур данных).

2.4. Примеры структур данных

Примеры предлагаемых структур данных:

- упорядоченный и неупорядоченный динамические массивы;
- упорядоченный список;
- бинарное дерево поиска;
- 2-3 дерево;
- сбалансированное дерево;
- красно-черное дерево;
- B-дерево;
- хэш-множество на основе одной из вышеперечисленных структур данных.

Возможно, имеет смысл давать студентам заготовку программы, так, чтобы им нужно было только написать структуры данных, реализующие заданный интерфейс.

3. Информационно-графовая модель данных

3.1. Теоретические упражнения

3.1.1. Понятие информационного графа

1. По аналогии с одномерной задачей интервального поиска приведите тип, описывающий n-мерные задачи интервального поиска.

2. Опишите тип, задающий задачи интервального поиска на n-мерном булевом кубе, которые состоят в поиске в конечном подмножестве n-мерного булевого куба всех тех точек, которые попадают в подкуб, задаваемый запросом. Какова мощность множества запросов у данного типа?

3. Задача о метрической близости состоит в том, чтобы по произвольно взятой точке-запросу единичного n-мерного куба n-мерного евклидового пространства найти

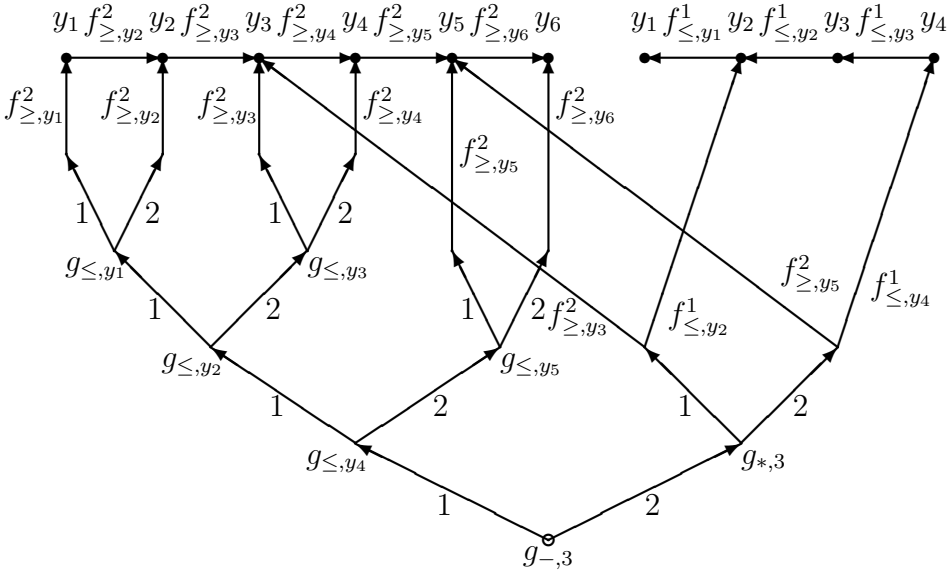

Рис. 1:

в конечном подмножестве этого куба (библиотеке) все точки, находящиеся на расстоянии не более, чем R от точки запроса. Опишите тип, задающий задачи о метрической близости.

4. Опишите тип, задающий задачи включающего поиска. Напомним, что в задаче включающего поиска имеется некоторое конечное множество свойств, и каждый элемент библиотеки (множества данных) обладает или не обладает каждым из этих свойств. Запрос задает некоторое подмножество множества свойств, и необходимо найти все элементы библиотеки, которые обладают всеми свойствами из запроса.

5. Рассмотрим следующую задачу поиска, которая может возникнуть, например, при разгадывании кроссвордов. Элементы библиотеки (записи) есть слова фиксированной длины n в алфавите $\{0,1\}$. Запрос задает некоторый набор позиций и значения букв в этих позициях. Необходимо найти в библиотеке все записи, у которых в позициях, задаваемых запросом, стоят буквы, совпадающие с соответствующими значениями позиций запроса. Опишите тип, задающий эти задачи поиска. Сравните полученный тип с типом задач интервального поиска на булевом кубе (см. упражнение 2).

3.1.2. Критерий допустимости информационных графов

6. Пусть $S = \langle X, X, = \rangle$ — тип поиска идентичных объектов, множество предикатов F задается соотношением

$$F = \{f_{=,a}(x) = \begin{cases} 0, & \text{если } x \neq a \\ 1, & \text{если } x = a \end{cases} : a \in X\}, \tag{1}$$

базовое множество имеет вид $\mathcal{F} = \langle F, \emptyset \rangle$, $V = \{y_1, y_2, \ldots, y_k\} \subseteq X$. Приведите пример информационного графа над базовым множеством $\mathcal{F}$, разрешающего ЗИП $I = \langle X, V, =$

$\rangle$.

7. Пусть $S = \langle X, X, = \rangle$ — тип поиска идентичных объектов, множество переключателей имеет вид

$$G = \{g_{=,a}(x) = \begin{cases} 1, & \text{если } x = a \\ 2, & \text{если } x \neq a \end{cases} : a \in X\}, \tag{2}$$

базовое множество имеет вид $\mathcal{F} = \langle \emptyset, G \rangle$, $V = \{y_1, y_2, \ldots, y_k\} \subseteq X$. Приведите пример информационного графа над базовым множеством $\mathcal{F}$, разрешающего ЗИП $I = \langle X, V, =$ $\rangle$.

8. Пусть $X = \{1, 2, ..., N\}$, $S = \langle X, X, = \rangle$ — тип поиска идентичных объектов, множество переключателей имеет вид

$$G = \{g_a(x) = \begin{cases} 1, & \text{если } x < a \\ 2, & \text{если } x = a \\ 3, & \text{если } x > a \end{cases} : a \in X\}, \tag{3}$$

базовое множество имеет вид $\mathcal{F} = \langle \emptyset, G \rangle$, $V = \{3, 5, 7, 11, 13, 17, 19\}$. Постройте информационный граф над базовым множеством $\mathcal{F}$, разрешающий ЗИП $I = \langle X, V, = \rangle$.

9. Пусть $X = \{1, 2, ..., N\}$, $S = \langle X, X, = \rangle$ — тип поиска идентичных объектов, $V = \{y_1, y_2, \ldots, y_k\} \subseteq X$. Предположим, что $y_1 < y_2 < \cdots < y_k$. Метод блочного поиска с размером блока m, решающий задачу $I = \langle X, V, = \rangle$, состоит в следующем. Если на вход алгоритма поиска подается запрос $x \in X$, то, начиная с $i = 1$ до $i = k/m$, просматриваем записи $y_{i \cdot m}$. Если $x > y_{i \cdot m}$, то увеличиваем i на 1, иначе по очереди просматриваем записи $y_{(i-1)m+1}, y_{(i-1)m+2}, \ldots, y_{i \cdot m}$ и сравниваем их с запросом x. При равенстве мы нашли нужную запись, если же ни для какой записи равенства не наблюдается, то ответ на запрос x пуст. Опишите базовое множество и постройте информационный граф над этим базовым множеством, который бы решал ЗИП $I = \langle X, V, = \rangle$ методом блочного поиска.

10. Пусть $X = \{1, 2, ..., N\}$, $V \subseteq X$, ρ_c — отношение поиска, задаваемое на $X \times V$ и определяемое соотношением

$$x \rho_c y \Longleftrightarrow (y \in V) \& (x \leq y) \& (\neg(\exists y')((y' \in V) \& (x \leq y') \& (y' < y))), \tag{4}$$

т.е. $x \rho_c y$, если $y \in V$, ближайшее справа к x. При выполнении этих условий ЗИП $I = \langle X, V, \rho_c \rangle$ называется задачей о близости. Пусть базовое множество имеет вид $\mathcal{F} = \langle \emptyset, G \rangle$, где множество переключателей G задается соотношением

$$G = \{g_{\leq,a}(x) = \begin{cases} 1, & \text{если } x \leq a \\ 2, & \text{если } x > a \end{cases} : a \in X\}. \tag{5}$$

Постройте информационный граф над базовым множеством $\mathcal{F}$, разрешающий ЗИП $I = \langle X, V, \rho_c \rangle$, если $V = \{3, 5, 7, 11, 13, 17, 19\}$.

11. Одномерная задача о доминировании задается типом $S_{dom1} = \langle [0, 1], [0, 1], \geq \rangle$. Пусть $V = \{y_1, y_2, \ldots, y_k\} \subseteq [0, 1]$. Опишите некоторое базовое множество и постройте какой-либо информационный граф над этим базовым множеством, который бы решал ЗИП $I = \langle [0, 1], V, \geq \rangle$.

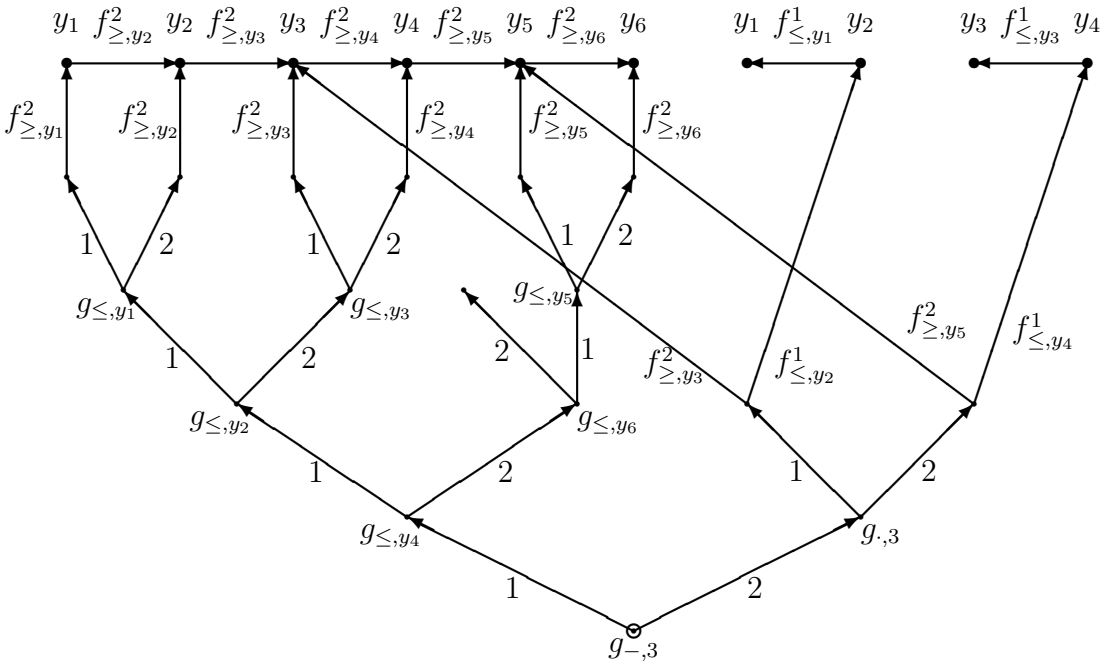

Рис. 2: Решение одномерной задачи интервального поиска

12. Пусть $S_{int} = \langle X_{int}, Y_{int}, \rho_{int} \rangle$ — тип одномерного интервального поиска, где отношение ρ_{int} определяется соотношением

$$(u, v)\rho_{int} y \Longleftrightarrow u \leq y \leq v, \tag{6}$$

где $(u, v) \in X,\ y \in Y$, $V = \{y_1, y_2, \ldots, y_6\}$, где $y_1 = 1/6$, $y_2 = 1/4$, $y_3 = 3/8$, $y_4 = 2/5$, $y_5 = 3/4$, $y_6 = 7/8$. Разрешает ли информационный граф, изображенный на рисунке 1, где функции определяются соотношениями

$$f^1_{\leq,a}(u, v) = \begin{cases} 1, & \text{если } u \leq a \\ 0, & \text{если } u > a \end{cases}, \tag{7}$$

$$f^2_{\geq,a}(u, v) = \begin{cases} 1, & \text{если } v \geq a \\ 0, & \text{если } v < a \end{cases}, \tag{8}$$

$$g_{\cdot,m}(u, v) = \max(1,]u \cdot m[), \tag{9}$$

$$g_{-,m}(u, v) = \begin{cases} 1, & \text{если } v - u < 1/m \\ 2, & \text{если } v - u \geq 1/m \end{cases}, \tag{10}$$

$$g_{\leq,a}(u, v) = \begin{cases} 1, & \text{если } u \leq a \\ 2, & \text{если } u > a \end{cases}, \tag{11}$$

задачу информационного поиска $I = \langle X_{int}, V, \rho_{int} \rangle$? Обоснуйте ответ.

13. Докажите, что информационный граф, изображенный на рисунке 2, разрешает одномерную задачу интервального поиска $I = \langle X_{int}, V, \rho_{int} \rangle$, где $V = \{y_1, y_2, y_3, y_4, y_5, y_6\}$ — библиотека, изображенная на рисунке 3.

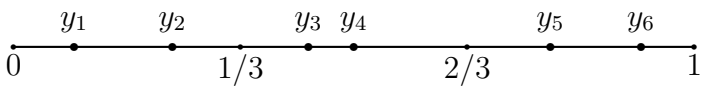

Рис. 3:

14. Пусть $S_{int} = \langle X_{int}, Y_{int}, \rho_{int} \rangle$ — тип одномерного интервального поиска, где отношение ρ_{int} определяется соотношением (6), $V = \{1/8, 1/7, 1/5, 3/7,\ 3/5, 4/5, 7/8\}$. Опишите некоторое базовое множество и постройте какой-либо информационный граф над этим базовым множеством, который бы решал ЗИП $I = \langle X_{int}, V, \rho_{int} \rangle$.

3.1.3. Полнота для информационных графов

15. Пусть $S = \langle X, X, = \rangle$ — тип поиска идентичных объектов, базовое множество имеет вид $\mathcal{F} = \langle \emptyset, G \rangle$, где множество переключателей G задается соотношением (5). Будет ли полно базовое множество $\mathcal{F}$ для типа S?

16. Задача включающего поиска, описанная в упражнении 4, может быть задана типом $S_{bool} = \langle B^n, B^n, \overset{b}{\succeq} \rangle$, где B^n — n-мерный булев куб, $\overset{b}{\succeq}$ — отношение поиска на $B^n \times B^n$, определяемое следующим соотношением

$$(x_1, \ldots, x_n) \overset{b}{\succeq} (y_1, \ldots, y_n) \Longleftrightarrow x_i \geq y_i,\ i = \overline{1, n}. \tag{12}$$

Приведите пример базового множества, полного для типа S_{bool}. Приведите пример минимального по мощности базового множества, полного для типа S_{bool}.

3.1.4. Сложность информационных графов

17. Пусть $X = \{1, 2, ..., N\}$, $S = \langle X, X, =, \mathbf{P}, \sigma \rangle$ — тип поиска идентичных объектов, где $\sigma = 2^X$, $\mathbf{P}$ — равномерная вероятностная мера, то есть для любого $x \in X$ выполняется $\mathbf{P}(x) = 1/N$.

1. Посчитайте сложность, B-сложность и объем информационного графа, полученного при решении упражнения 6.

2. Посчитайте сложность, B-сложность и объем информационного графа, полученного при решении упражнения 7.

3. Посчитайте сложность, B-сложность и объем информационного графа, полученного при решении упражнения 8. Для базового множества и ЗИП, приведенных в упражнении 8, постройте информационный граф со сложностью, не большей, чем 1.48.

4. Посчитайте сложность, B-сложность и объем информационного графа, полученного при решении упражнения 9, если $N = 100$. Для какого значения параметра m сложность будет минимальна. Для какого значения параметра m B-сложность будет минимальна. Для какого значения параметра m объем будет минимальным.

18. Если $X = \{1, 2, ..., N\}$ и на X задана равномерная вероятностная мера, то посчитайте сложность, B-сложность и объем информационного графа, полученного при решении упражнения 10.

19. Пусть на множестве запросов $X = [0, 1]$ задана равномерная вероятностная мера. Посчитайте сложность, B-сложность и объем информационного графа, полученного при решении упражнения 11.

20. Пусть на множестве запросов $X_{int} = \{(u, v) : 0 \le u \le v \le 1\}$ задана равномерная вероятностная мера. Посчитайте сложность, B-сложность и объем информационного графа, изображенного на рисунке 2.

3.1.5. Мощностная нижняя оценка

21. Пусть $X = \{1, 2, ..., N\}$, $S = \langle X, X, =, \mathbf{P}, \sigma\rangle$ — тип поиска идентичных объектов, где $\sigma = 2^X$, $\mathbf{P}$ — равномерная вероятностная мера, то есть для любого $x \in X$ выполняется $\mathbf{P}(x) = 1/N$, $V = \{3, 5, 7, 11, 13, 17, 19\}$. Приведите мощностную нижнюю оценку для ЗИП $I = \langle X, V, =\rangle$.

22. Пусть $S_{dom1} = \langle [0, 1], [0, 1], \ge, \mathbf{P}, \sigma\rangle$ — тип одномерной задачи о доминировании, где $\mathbf{P}$ — равномерная вероятностная мера на $[0, 1]$, $V = \{y_1, y_2, \ldots, y_k\} \subseteq [0, 1]$. Приведите мощностную нижнюю оценку для ЗИП $I = \langle [0, 1], V, \ge\rangle$.

23. Пусть $S_{int} = \langle X_{int}, Y_{int}, \rho_{int}\rangle$ — тип одномерного интервального поиска и на множестве запросов $X_{int} = \{(u, v) : 0 \le u \le v \le 1\}$ задана равномерная вероятностная мера. $V = \{y_1, y_2, \ldots, y_k\} \subseteq [0, 1]$. Приведите мощностную нижнюю оценку для ЗИП $I = \langle X_{int}, V, \rho_{int}\rangle$. Оцените сверху полученную величину.

24. Пусть $V = \{y_1, y_2, \ldots, y_k\} \subseteq B^n$ и число единиц в наборе y_i равно t_i ($i = 1, 2, \ldots, k$). Приведите мощностную нижнюю оценку для задачи включающего поиска $I = \langle B^n, V, \overset{b}{\succeq}\rangle$.

3.2. Задачи “реальный сценарий”

3.2.1. Краткое описание

Студенту предлагается решить какую-нибудь реальную прикладную задачу, для которой он должен сделать три вещи (предполагается, что он уже знаком с информационно-графовой моделью поиска, если нет, то сначала почитать соответствующую литературу):

- Сформулировать задачу в терминах информационно-графовой модели

  - определить множество элементов базы данных Y
  - определить понятие библиотеки V
  - определить множество запросов X
  - определить отношение равенства $\rho(x, y), x \in X, y \in Y$

- для произвольной базы данных привести алгоритм построения графа, решающего задачу поиска в этой базе
  - построить граф для какой-нибудь конкретной базы данных
  - вычислить сложность этого графа (временную T и объемную Q)
  - посчитать (или оценить) сложность графа для поставленной задачи и произвольной базы данных
  - (optional) сложность должна удовлетворять заданным условиям

- реализовать алгоритм решения поставленной задачи на компьютере
  - алгоритм должен использовать информационный граф, построенный в предыдущем пункте
  - проверить на входных данных корректность работы алгоритма
  - проверить на входных данных оценки сложности, полученные в предыдущем пункте

3.2.2. Примеры задач "реальный сценарий"

Русско-англо-русский словарь. Задача представляет из себя написание словаря, который на ввод пользователем слова на одном из языков выводит все возможные переводы на другой язык. В качестве языков могут быть выбраны как реальные языки (английский, русский), так и абстрактные (A^+, B^+).

Решение задачи предполагает написание конструкции, которая принимает пары $(a, b) \in A \times B$ и заполняет ими базу данных. Конструкция должна обладать свойством быстрого доступа, т.е. должна быстро выдавать ответы на запрос. Для этого хорошо подходит, например, хэш-дерево, ключем в котором является первая буква слова (две буквы слова, если букв в алфавите мало), потом вторая буква, если это необходимо, и т.д.

Интернет-магазин. Предположим, у нас есть интернет-магазин, продающий товары одной категории, который представляет возможность пользователю запросить параметры товара и выбрать из подходящих наиболее интересный.

Задача представляет из себя поиск в базе данных объектов, которые удовлетворяют заданным критериям. А именно, каждый товар есть, по сути, набор атрибутов $(a_1, a_2, \ldots, a_s) \in A_1 \times A_2 \times \ldots \times A_s$, где $A_1, A_2, \ldots, A_s$ — некоторые

множества, такие как $\{0,1\}$, отрезок натурального ряда $1,\ldots,r_i$ или отрезок вещественных чисел $[0,1]$. Запрос также представляет из себя вектор длины s из множества $(A_1 \vee \{*\}) \times (A_2 \vee \{*\}) \times \ldots \times (A_s \times A_s \vee \{*\})$, где $*$ означает, что пользователю не интересен данный параметр, а для остальных это либо значение из A_t, которое должен принимать параметр k товаров (для булевых и целых параметров), или интервал из $A_t \times A_t$, в котором должен лежать параметр k товара (для натуральных или вещественных параметров).

Граф, решающий данную задачу должен представлять из себя конструкцию, последовательно проверяющую параметры запроса и отсекающую неподходящие варианты товара. Каждая функция должна учитывать дополнительный вариант значения запроса $*$.

3.3. Моделирование информационных графов

3.3.1. Создание класса “информационный граф”

Задача состоит в написании класса (**class**), моделирующего информационный граф. Основной метод класса — это метод поиска. Входным данным метода является запрос, а выходными данными — множество записей, являющихся ответом информационного графа на запрос и целое число, равное количеству вычисленных предикатов и переключателей на данном запросе.

3.3.2. Реализация алгоритма поиска

Задается некоторый тип задач информационного поиска. Необходимо реализовать две процедуры.

Первая процедура получает на вход библиотеку (множество записей). В результате работы процедуры должен быть построен информационный граф, задаваемый описанным в предыдущем разделе классом и решающий заданную задачу информационного поиска.

Вторая процедура основывается на методе поиска, реализованного в предыдущем разделе. На вход процедуре поступает поток запросов на поиск, на выходе получается поток ответов (подмножеств библиотеки) и количество вычисленных функций в среднем на потоке.

3.3.3. Типы задач поиска

Предлагаемые к реализации типы задач поиска и алгоритмы их решения:

- задача о близости в линейно упорядоченном множестве и алгоритм решения с использованием хэш-функции;

- задача об одномерной метрической близости и оптимальный алгоритм ее решения на основе построения опорного множества;
- двумерная задача интервального поиска при условии, что каждый интервал-запрос представляет собой квадрат фиксированного размера и алгоритм решения этой задачи, основывающийся на методе сеток.