

КОМПЬЮТЕРНАЯ ДИСКРЕТНАЯ МАТЕМАТИКА

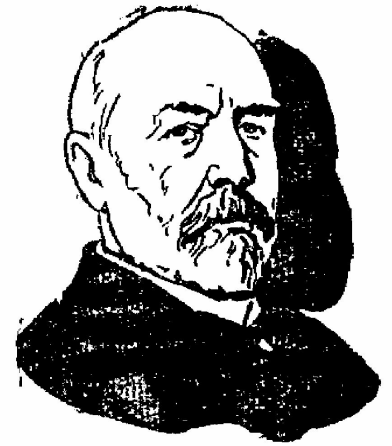
ПРЕДСТАВЛЕНИЕ МНОЖЕСТ В ЭВМ

ВО

ЛЕКЦИЯ 1

Историческая справка

- **Немецкий ученый, математик, создатель теории множеств**
- **Родился в Петербурге в 1845г.**
- **В 1867 г. окончил Берлинский университет**
- **В 1872-1913 гг. – профессор университета в Галле**
- **Сформулировал общее понятие мощности множества (1878)**
- **Развил принципы сравнения мощностей множеств и**
- **Систематически изложил принципы своего учения**
- **Созданная Кантором теория множеств, некоторые идеи которой имелись у его предшественников, послужила причиной общего пересмотра логических основ математики и оказала влияние на всю современную ее структуру.**



**Георг Кантор
(XIX-XXвв.)**

Объекты изучения: множество, элемент множества, операции над множествами

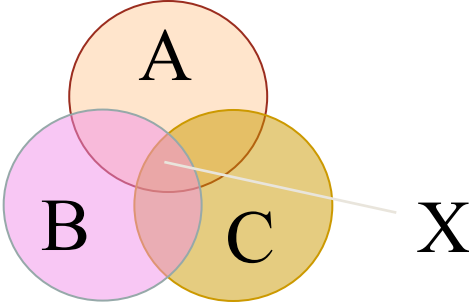
Цели:

- 1) описать способы хранения информации о принадлежности элемента множеству,
- 2) описать алгоритмы для вычисления результатов операций над множествами

Один и тот же объект может быть представлен разными способами. Выбор представления зависит от многих факторов: особенностей представляемого объекта, частоты его использования в задаче и т. д. Программисту надо знать много способов и уметь выбрать наиболее подходящий для рассматриваемой задачи.

Некоторые способы задания множеств



Способ	Пример
Перечислением элементов	$\{a,b,c\}$, $A=\{1,3,5,7\}$
Характеристическим свойством $M=\{x \mid P(x)\}$ P – характеристическое свойство элементов x	$M=\{x \mid x=2k, k \in \mathbb{N}\}$ - множество четных натуральных чисел; $M=\{x \mid \sin x = 1\}$ - множество решений уравнения $\sin x = 1$.
Порождающей процедурой (например, операции над множествами)	$X=(A \cup B) \cap C$
Графически при помощи диаграмм Эйлера	

Булеан. Мощность булеана

- **Булеан** – множество всех подмножеств данного множества M
- Обозначение: $V(M)$
- **Пример:** дано множество $A=\{a,b,c\}$. Найти $V(A)$.

$$V(A)=\{ \emptyset, \{a\}, \{b\}, \{c\}, \{a,b\}, \{a,c\}, \{b,c\}, \{a,b,c\} \}$$

- **Мощность булеана** определяется по формуле:
$$|V(M)|=2^{|M|}$$
- Пустое множество и само множество являются **несобственными** подмножествами множества M
- Остальные подмножества – **собственные**





Операции над множествами

Название операции	Определение	Диаграммы Эйлера
Пересечение	$A \cap B = \{ x \mid x \in A \text{ и } x \in B \}$	
Объединение	$A \cup B = \{ x \mid x \in A \text{ или } x \in B \}$	
Разность	$A \setminus B = \{ x \mid x \in A \text{ и } x \notin B \}$	
Дополнение	$\bar{A} = U \setminus A = \{ x \mid x \in U \text{ и } x \notin A \}$	
Симметрическая разность	$A \Delta B = (A \setminus B) \cup (B \setminus A)$	

МАШИННЫМ СЛОВОМ ИЛИ БИТОВОЙ ШКАЛОЙ

Пусть универсальное множество конечно и число n его элементов не превосходит разрядности ЭВМ. Занумеруем элементы универсума, т. е. . Подмножество A универсума представляется кодом C , в котором

где
$$C[i] = \begin{cases} 1, & \text{если } u_i \in A, \\ 0, & \text{если } u_i \notin A. \end{cases}$$
 - это i -тый разряд кода C . Тогда, код пересечения множеств A и B есть поразрядное логическое произведение кода множества A и кода множества B . Код объединения множеств A и B есть поразрядная логическая сумма кода множества A и кода множества B . Код дополнения множеств A до универсума есть инверсия кода множества A . в большинстве ЭВМ для этих операций есть соответствующие команды. Таким образом, операции над небольшими множествами выполняются эффективно.

Представление множества упорядоченными списками

- Если универсум очень велик или бесконечен, а рассматриваемые подмножества не очень велики, то представление с помощью битовых шкал не является эффективным с точки зрения экономии памяти. В этом случае множества представляют списками элементов, часто упорядоченными.
- Эффективная реализация операций над множествами в этом случае основана на общем алгоритме, известном под названием «Алгоритм типа слияния».
- Такой алгоритм параллельно просматривает два множества, представленных упорядоченными списками, причем на каждом шаге продвижение происходит в том множестве, в котором текущий элемент меньше.

Алгоритм нахождения пересечения слиянием

- Множества будем хранить как массив с нумерацией элементов, начинающейся с единицы.
- Обозначим через i номер текущего рассматриваемого элемента в множестве A , через j – номер текущего рассматриваемого элемента множества B . Будем получать множество P , представляющее собой пересечение множеств A и B . Через k обозначим мощность множества P . Также k будет и номером последнего добавленного элемента в P

Алгоритм решения

- Положить $i = j = 1$ и $k = 0$.
 - Если в A и B (одновременно) есть ещё не просмотренные элементы, выполнить:
 - Если $A[i] = B[j]$, то выполнить:
 - i. Добавить $A[i]$ в P , то есть $k := k + 1$ и $P[k] := A[i]$
 - ii. Перейти к следующим элементам множеств A , B , то есть $i := i + 1$ и $j := j + 1$
 - Если $A[i] < B[j]$, то перейти к следующему элементу множества A , то есть $i := i + 1$
 - В остальных случаях (то есть когда $A[i] > B[j]$) перейти к следующему элементу множества B , то есть $j := j + 1$
 - Перейти к пункту 2.
 - **Пример.** $A = \{1,3,6,8,9\}$ $B = \{3,4,5,6,7\}$
 -
 - Сравниваемые пары: $(1,3)$, $(3,3)$, $(6,4)$, $(6,5)$, $(6,6)$, $(8,7)$
 -
- P 3 6

Алгоритм нахождения объединения слиянием

Обозначим через i номер текущего рассматриваемого элемента в множестве A , через j – номер текущего рассматриваемого элемента множества B . Будем получать множество P , представляющее собой объединение множеств A и B . Через k обозначим мощность множества P . Также k будет и номером последнего добавленного элемента в P .

Алгоритм решения.

Положить $i = j = 1, k = 0$.

Если ещё не просмотрены все элементы множеств A, B выполнить:

Если в A ещё есть элементы, и в B есть элементы и $A[i] = B[j]$, то

- i. Добавить $A[i]$ в P , то есть $k := k + 1$ и $P[k] := A[i]$
- ii. Перейти к следующим элементам в A и B , то есть $i := i + 1$ и $j := j + 1$
Если в B уже все элементы были просмотрены или же $A[i] < B[j]$ (при условии, что в A не все элементы были просмотрены) выполнить:
- iii. Добавить $A[i]$ в P , то есть $k := k + 1$ и $P[k] := A[i]$
- iv. Перейти к следующему элементу множества A , то есть $i := i + 1$

Во всех остальных случаях (то есть когда в A уже все элементы просмотрены или же если $A[i] > B[j]$) выполнить:

- i. Добавить $B[j]$ в P , то есть $k := k + 1$ и $P[k] := B[j]$;
- ii. Перейти к следующему элементу множества B , то есть $j := j + 1$
Перейти к пункту 2.

Как видно, на каждом шаге мы добавляем в P минимальный элемент из $A[i]$ и $B[j]$ и переходим к рассмотрению следующего элемента.

Пример. $A = \{1,3,6,8,9\}, B = \{3,4,5,6,7\}$

Сравниваемые пары: $(1,3), (3,3), (6,4), (6,5), (6,6), (8,7)$

P 1 3 4 5 6 7 8 9