

# Лекция 2

## МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САПР

### Основные понятия и определения

- 1 Требования к математическому обеспечению САПР ЭА
- 2 Методы повышения эффективности САПР ЭА
- 3 Основы теории графов и их применение в ИТАП ЭА
- 4 Основы теории алгоритмов
- 5 Способы записи алгоритмов

# Вопрос 1 Требования к математическому обеспечению САПР ЭА

# Требования к МО САПР

## МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ -

совокупность математических методов и моделей алгоритмов проектирования, необходимых для их выполнения.



# Требования к МО САПР

1. **Специальная часть** – отображает специфику объекта проектирования, физические и информационные особенности его функционирования.

Специальная часть тесно привязана к этапам проектирования.

2. **Инвариантная часть** - включает методы и алгоритмы, слабосвязанные с особенностями математических моделей и используется на различных этапах проектирования.

# Требования к МО САПР

## 1. Универсальность -

применимость к широкому классу проектируемых объектов (нет количественной оценки)

## 2. Алгоритмическая надежность

свойство компонента МО давать при его применении правильные результаты (Количественная оценка - вероятность получения правильных результатов при соблюдении оговоренных ограничений на применение метода)

# Требования к МО САПР

## 3. Точность

определяется по степени совпадения расчетных и истинных результатов (при решении тестовых задач)

## 4. Затраты машинного времени (min)

главный ограничивающий фактор при повышении сложности проектируемых в САПР объектов

## 5. Используемая память (min)

## Вопрос 2 Методы повышения эффективности САПР ЭА

# Методы повышения эффективности САПР

1. Разработка экономических моделей и алгоритмов, имеющих частный характер
2. Совершенствование используемых **общих принципов**, создание МО эффективного по затратам машинного времени и памяти.



# Общие принципы создания МО САПР ЭА

1. Учет разряженности матриц
2. Исследование сложных систем по частям
3. Макромоделирование
4. Событийность анализа
5. Рациональное использование эвристических способностей человека в интерактивных процедурах

# Вопрос 3 Основы теории графов и их применение в ИТАП ЭА

# Основные определения

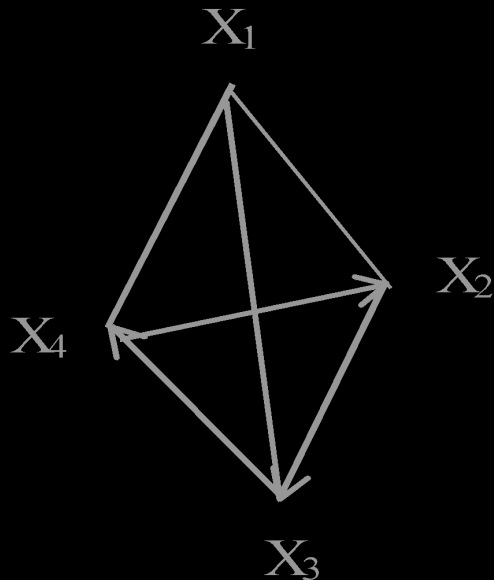
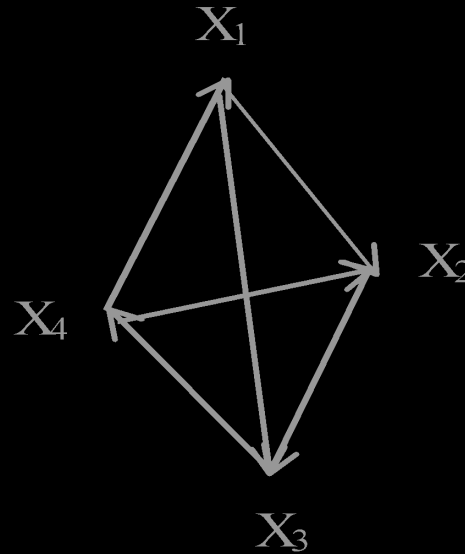
Под графом  $G(X, U)$  понимают совокупность непустого множества  $X$  и изолированное от него подмножество  $U$ , возможно нулевое, представляющее собой множество всех упорядоченных пар  $x_i x_j$ , где  $x_i, x_j$  принадлежат  $X$ ;  $i, j = 1 \dots n$ , где  $n$  – мощность множества.

Элементы множества  $X$  и  $U$  соответственно называются **вершинами** и **ребрами** графа

# Основные определения

Виды графов:

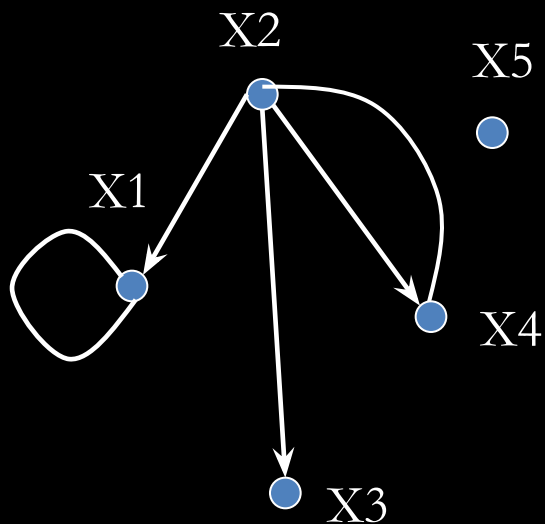
1. Неориентированные
2. Ориентированные →
3. Смешанные



Граф  $G(X, U)$  называется **неориентированным**, если для каждого его ребра несущественен порядок двух его концевых вершин.

# Основные определения

- 1) Граф, у которого 2 вершины соединены более чем одним ребром – **мультиграф**.
- 2) Ребра, у которых обе концевые вершины совпадают, называются **петлями**.
- 3) Вершина неинцидентная никакому ребру называется **изолированной**.



# Основные определения

Число ребер инцидентных некоторой вершине  $x_i$  называется **степенью вершины**.

Граф, состоящий только из изолированных вершин называется **нуль-графом**.

Граф **конечен**, если содержит конечное число вершин и ребер.

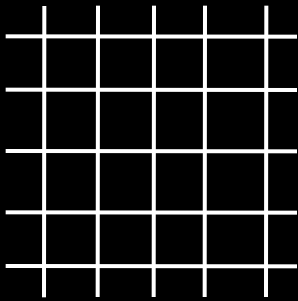
Конечный граф, у которого отсутствуют петли и изолированные вершины называется **регулярным**.

# Основные определения

Граф называется **однородным степени  $t$** , если степень всех его вершин  $= t$ .

Граф, все вершины которого попарно смежны называется **полным графом**.

Полный граф, у которого при каждой вершине имеется петля, называется **плотным**.



$$t = 4$$

# Основные определения

Граф, в котором перемещаясь по ребрам из вершины в вершину можно попасть в каждую вершину называется **связным графом**.

Число, характеризующее разность между числом верши графа (**мощностью**)  $n$  и числом компонент связности  $p$  называют **рангом графа** ( $R(G)$ ).

Один и тот же граф может иметь различные геометрические **реализации** (**изоморфные графы**).



# Основные определения

**Циклом** называется последовательность ребер, при которой в результате обхода вершин графа по этим ребрам возвращаются в исходную вершину.

Последовательность ребер при переходе от одной вершины к другой называется **цепью**.

**Эйлеров цикл** – это цикл, в котором содержатся все ребра графа.

# Основные определения

Цикл называют **Гамильтоновым**, если он проходит через каждую вершину графа только один раз.

Связной неориентированный граф, не содержащий циклов, называется **деревом**.

Несвязной граф без циклов, отдельные компоненты связности которого являются деревьями называется **лесом**.

Под **расстоянием между вершинами графа** понимается длина кратчайшей цепи, соединяющей эти вершины.

**Диаметр графа** – это максимальное расстояние между вершинами графа.

# Основные определения

Объект  $H(X, E)$  считается гиперграфом, если он состоит из множества вершин  $X$  и множества ребер  $E$ , причем каждое ребро  $e_i$ , принадлежащее  $E$  является некоторым подмножеством множества  $X$ . Т.е. множество  $X$  должно включать любое ребро  $e_i$ .

При этом каждое ребро может соединять не только две вершины, но и любое подмножество множества вершин графа.

# Вопрос 4 Основы теории алгоритмов

# Основы теории алгоритмов

**Алгоритм** – это конечная совокупность точно заданных правил решения произвольного класса задач.

Алгоритм состоит из отдельных конечных действий – **шагов**.

## Универсальные алгоритмические модели

1. Понятие алгоритма связывается с вычислениями и числовыми функциями.
2. Алгоритм представляется как некоторое детерминированное устройство, способное выполнять в определенные моменты времени типовые (простейшие) операции.
3. Производится преобразование слов произвольных алфавитов.

# Основы теории алгоритмов

**Детерминированный алгоритм** - если он выражен системой правил, однозначно определяющих результат процесса при заданных исходных данных.

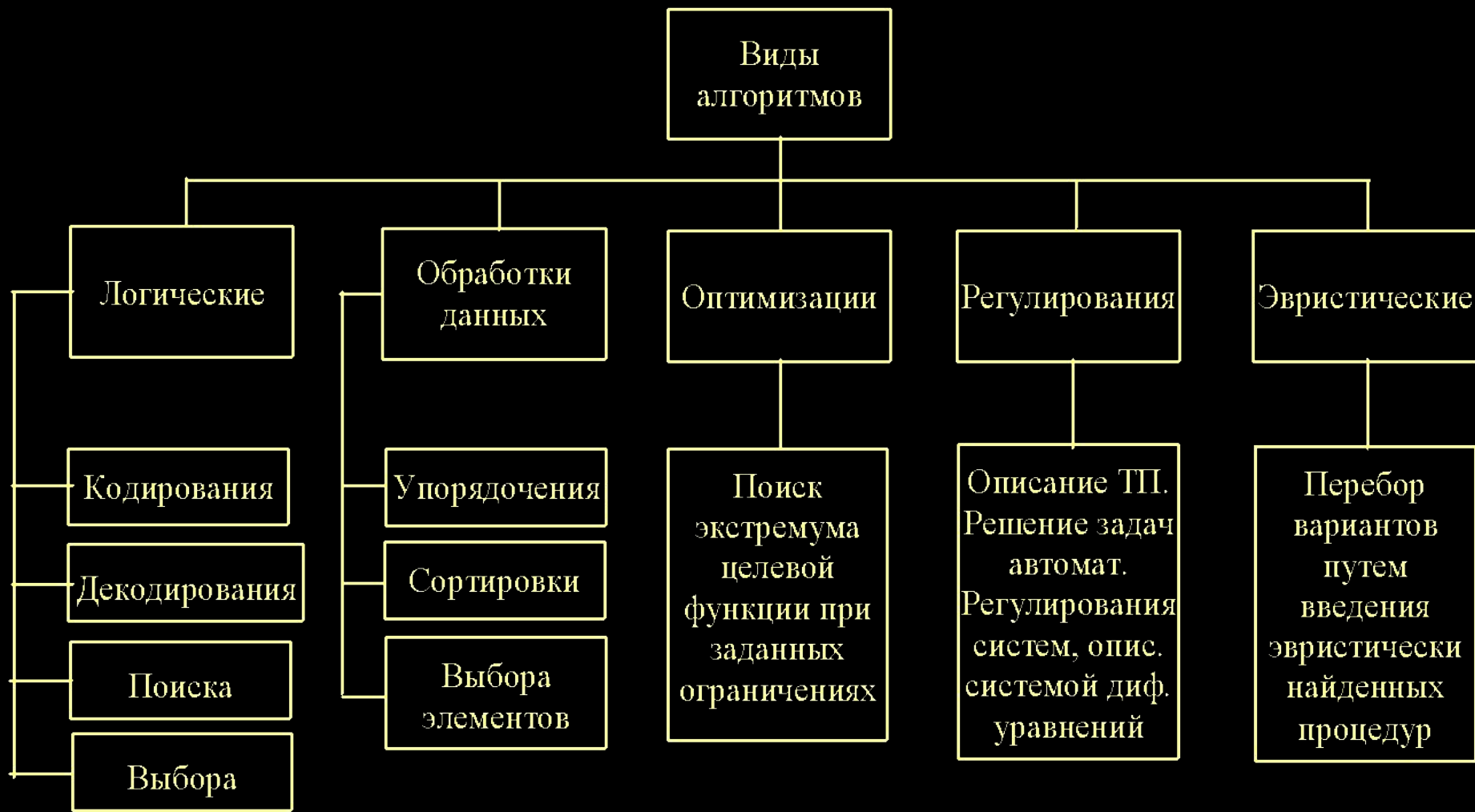
Если правила неоднозначны и результаты можно представить только статистически, то такой алгоритм называется **вероятностным** или **стахостическим**.

Если правило нельзя задать ни вероятностно, ни детерминированно, но можно сформировать содержательные указания о целенаправленности процесса, то такой алгоритм называется **эвристическим**.

# Основы теории алгоритмов

Расшифровка укрупненных операторов алгоритма в командах языка компьютера называется **программированием**, а запись алгоритма на входном языке компьютера - **программой**.

# Классификация алгоритмов при проектировании ЭА





# Свойства алгоритмов

1. **Массовость** – свойство алгоритма отображать широкий класс процессов.
2. **Результативность** - свойство алгоритма обеспечивать получение результата через конечное число шагов.
3. **Область применения** – множество процессов, для которых алгоритм результативен.
4. **Определимость** - свойство алгоритма, заключающееся в том, что каждый его шаг определяется точно.

# Свойства алгоритмов

5. Алгоритм имеет **вход** и **выход**.
6. Алгоритмы являются **эквивалентными** – если совпадают их области применимости и результаты обработки любого процесса из данной области.
7. Алгоритмы **равны** – если равны соответствующие им операторы и совпадают системы правил, задающие действия этих алгоритмов.

# Эффективность алгоритмов

**Эффективность** – оценка максимального числа элементарных операций, выполняемых при работе алгоритма.

## Численная оценка эффективности

1. Общее время реализации алгоритма:

$$T = \sum_{i=1}^n Q_i t_i$$

где  $Q_i$  – число операций  $i$ -го типа,  $n$  – число типов операций в алгоритме,  $t_i$  – время выполнения  $i$ -й операции.

# Эффективность алгоритмов

2. Общее число операций, приведенных к элементарной :

$$N_{\text{э}} = \sum_{i=1}^n B_i Q_i$$

$$B_i = \frac{t_i}{t_{\text{э}}}$$

где  $t_{\text{э}}$  – время выполнения элементарной операции

Тогда общее время реализации алгоритма

$$T = N_{\text{э}} \cdot t_{\text{э}}$$

# Эффективность алгоритмов

3. Объем памяти, которую необходимо зарезервировать в компьютере для реализации алгоритма:

$$V = V_{П} + V_{ВХ} + V_{ПР} + V_{ВЫХ}$$

где  $V_{П}$  – объем памяти для размещения программы,  $V_{ВХ}$ ,  $V_{ВЫХ}$  – объем памяти для размещения исходной входной и выходной информации,  $V_{ПР}$  – объем памяти, необходимой для размещения промежуточной информации.

Коэффициент сложности алгоритма тем выше, чем выше  $N_{Э}$  и  $V$ .

# Вопрос 5 Способы записи алгоритмов

# Способы записи алгоритмов

## 1) Операторный алгоритм Ван-Хао

Алгоритм задается последовательностью приказов специального вида. Каждый приказ имеет определенный номер и содержит следующие указания: **какую операцию следует выполнить над заданным объектом и приказ с каким номером следует далее выполнить действие над результатом данной операции.**

Общий вид приказа:

$$i : \omega \quad \alpha \quad \beta$$

где  $i$  – номер приказа,  $\omega$  - элементарная операция над объектом,  $\alpha$  и  $\beta$  - номера некоторых приказов

# Способы записи алгоритмов

## 1) Операторный алгоритм Ван-Хао

Выполнить приказ  $i$  над числом  $X$  в операторе алгоритма – значит найти число  $W(X)$  и затем перейти к выполнению приказа  $\alpha$ . Если значение  $W(X)$  неопределено, то перейти к выполнению над числом  $X$  приказа с номером  $\beta$ .



# Способы записи алгоритмов

## 2) Логическая схема алгоритма

логической схемой алгоритма называются выражения, составленные из операторов и логических условий, следующих один за другим. После каждого логического условия ставится стрелка, которая оканчивается у какого-либо оператора (↑ начало, ↓ конец)

Для записи алгоритмов используют **основные типы операторов:**

- арифметические операторы (обозначаются начальными заглавными буквами латинского алфавита),
- операторы проверки логических условий (малые буквы латинского алфавита),
- операторы переадресации (обозначается буквой F (..). В скобках указан изменяемый адрес или параметр),
- операторы переноса,
- операторы формирования.

# Способы записи алгоритмов

## Логическая схема алгоритма

Пример:  $A_0 p_1 \uparrow_1 A_1 \downarrow_1 p_2 \uparrow_2 A_2 A_3 \downarrow_2 A_4 A_k$

$A_0$  ,  $A_k$  – операторы начала и конца,  $A_1 \dots A_4$  – операторы,  $p_1$  ,  $p_2$  – логические условия.

*Если  $p_1 = 1$ , то произойдет переход на оператор  $A_1$ , если  $= 0$ , то произойдет переход на оператор  $p_2$*

# Способы записи алгоритмов

## 3) Структурная схема алгоритма

Алгоритм расчленяется на отдельные блоки, которые отображаются в виде геометрических фигур. Блоки нумеруются и внутри них указываются действия, которые записываются в виде формул или словестно.

Блоки соединяются стрелками, показывающими связи между ними.

Если логические условия передают управления другим блокам, то на стрелках этих блоков указываются условия, при которых процесс разветвляется.

# Способы записи алгоритмов

## Структурная схема алгоритма

### Достоинства:

1. Обеспечивается возможность обмена структурными схемами алгоритмов между специалистами.
2. Обеспечивается наглядное чтение и понимание алгоритмов.
3. Уменьшается число ошибок при программировании.

# Способы записи алгоритмов

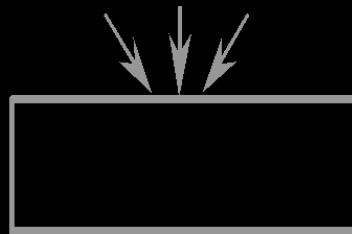
## Структурная схема алгоритма. Примеры записи вершин графа



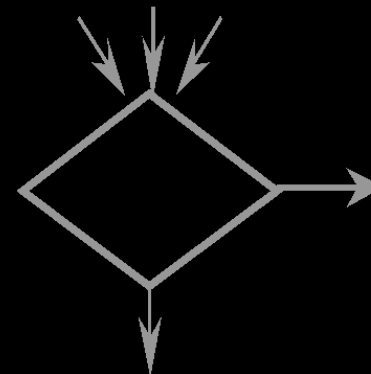
Н а ч а л ь н а я  
в е р ш и н а



К о н е ч н а я  
в е р ш и н а

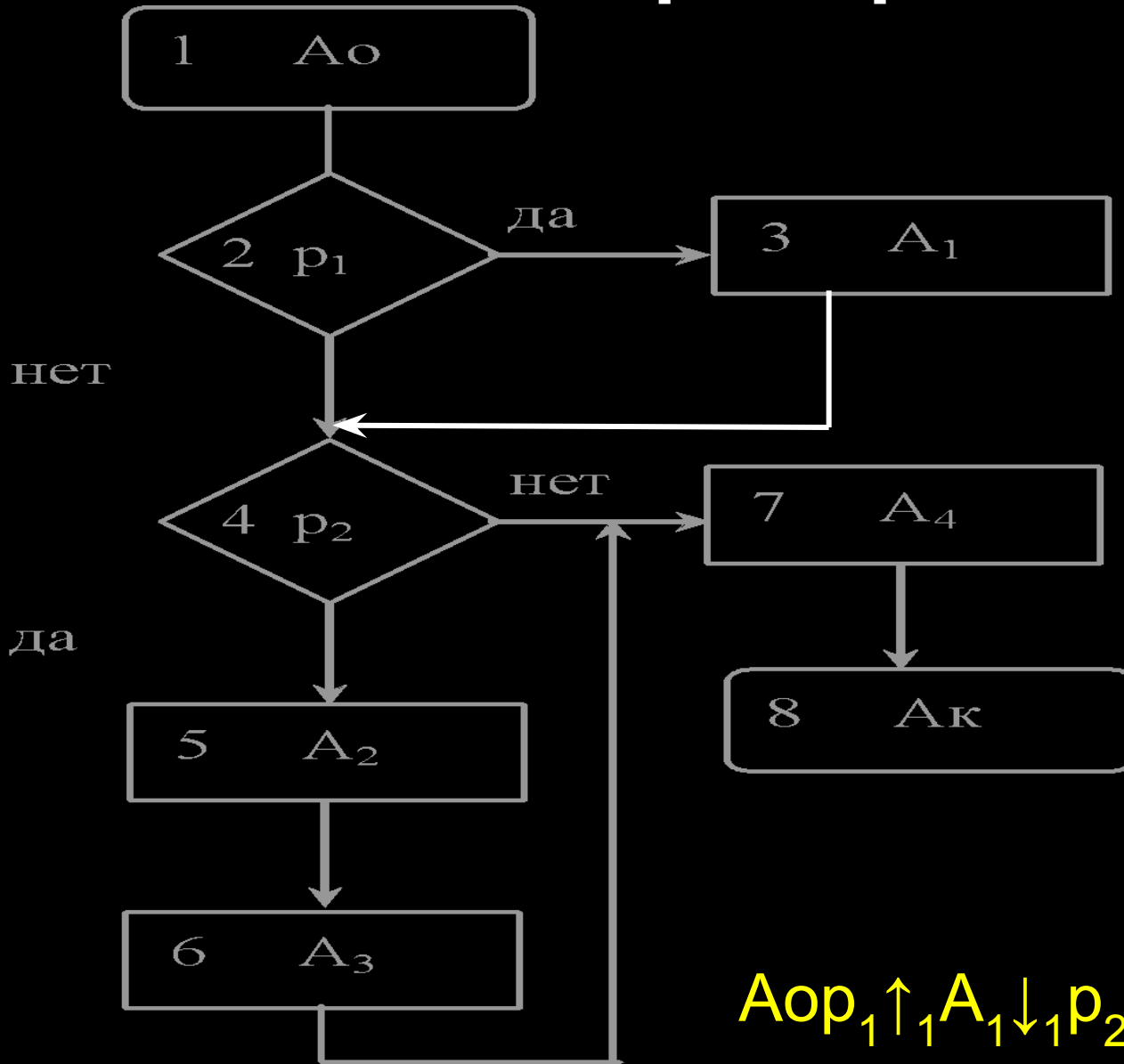


О п е р а т о р н а я  
в е р ш и н а



У с л о в н а я  
в е р ш и н а

# Структурная схема алгоритма. Пример



$A_0 \rightarrow p_1 \uparrow_1 \rightarrow A_1 \downarrow_1 \rightarrow p_2 \uparrow_2 \rightarrow A_2 \rightarrow A_3 \downarrow_2 \rightarrow A_4 \rightarrow A_k$

# Способы записи алгоритмов

## 4) Словесное задание алгоритмов

При данном способе задания алгоритма перечисляются блоки алгоритма и в них записываются логические условия.

*Вопросы по прочитанному  
материалу?*



*Спасибо за внимание!*