Тишин В.В.

Отношение линейного порядка.

Отношение называется *отношением линейного порядка*, если оно - отношение частичного порядка и обладает связностью.
Обозначение $x \preccurlyeq y$ будем употреблять и для элементов, вступающих в отношение линейного порядка.

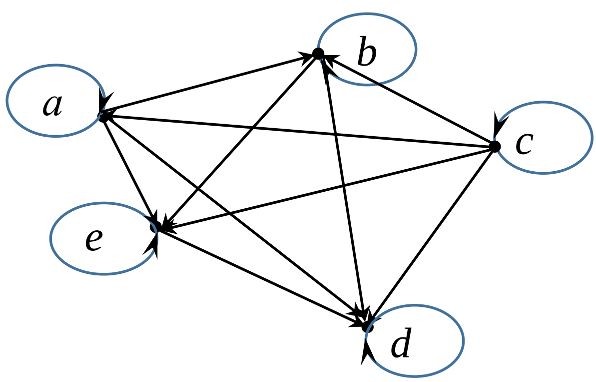
Примеры.

1. $A = R$, $x \preccurlyeq y \Leftrightarrow x \geq y$.
2. $A = C$, $(a,b) \preccurlyeq (c,d) \Leftrightarrow \left[\begin{array}{l} a < c \\ a = c, b \leq d \end{array}\right.$.
3. A – множество жителей Самары, $x \preccurlyeq y \Leftrightarrow x$ не моложе y.

Задание для самостоятельной работы:
Проверить для каждого из перечисленных отношений выполнение рефлексивности, антисимметричности, транзитивности и связности.

Граф отношения линейного порядка имеет у каждой вершины петлю, не содержит обоюдоострых стрелок, является транзитивным, и любые две его вершины соединены дугой.

Пример:



(рис.34)

Отношение строгого порядка.

Отношение называется *отношением строгого порядка*, если оно антирефлексивно, антисимметрично и транзитивно.
Если пара $(x;y)$ принадлежит графику отношения строгого порядка, то будем писать $x \prec y$.

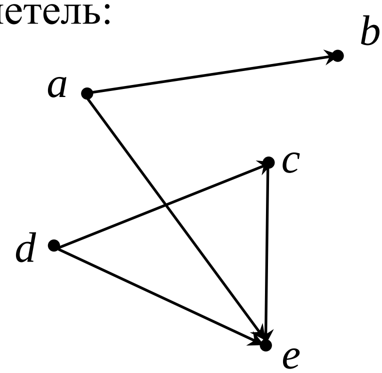
Примеры.

1. $A = U$, $X \prec Y \Leftrightarrow Y \subset X$.

2. $A=R,\ x \prec y \Leftrightarrow x<y.$
3. $A-$ множество жителей Самары, $x \prec y \Leftrightarrow x$ предок y.

Задание для самостоятельной работы:
Проверить для каждого из перечисленных отношений выполнение антирефлексивности, антисимметричности и транзитивности.

Граф отношения строгого порядка транзитивен, не содержит обоюдоострых дуг и петель:



(рис.35)

Отношение строгого линейного порядка.

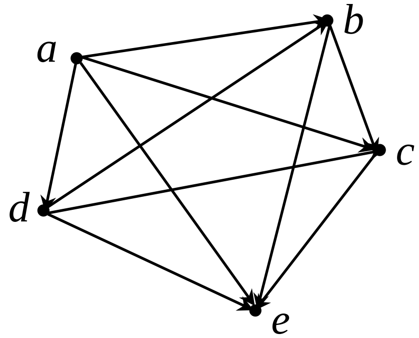
Отношение называется *отношением строгого линейного порядка*, если оно – связное отношение строгого порядка.
Если пара $(x;y)$ принадлежит графику отношения строгого линейного порядка, то, как и в случае отношения строгого порядка, будем писать $x \prec y$.

Примеры.

1. $A=R,\ x \prec y \Leftrightarrow x<y.$
2. $A=C,\ (a,b) \prec (c,d) \Leftrightarrow \left[\begin{array}{l} a>c \\ a=c,\ b>d \end{array}\right..$
3. $A-$ множество слов русского языка, $x \prec y \Leftrightarrow$ слово x расположено в энциклопедическом словаре раньше слова y.

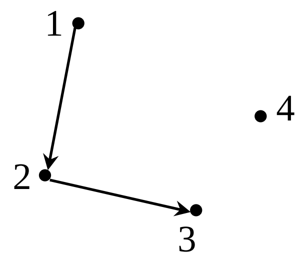
Задание для самостоятельной работы:
Проверить для каждого из перечисленных отношений выполнение антирефлексивности, антисимметричности, транзитивности и связности.

Граф отношения строгого линейного порядка транзитивен, связен и не содержит обоюдоострых дуг и петель:



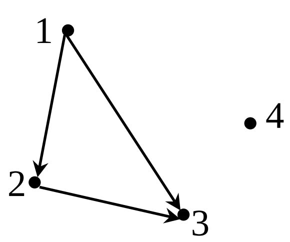
(рис.36)

Пример: Добавляя наименьшее возможное число рёбер к графу, изображённому на рисунке, добиться, чтобы этот граф стал графиком отношения

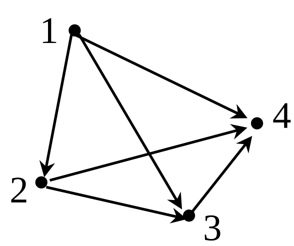


а) строгого порядка;
б) строгого линейного порядка;
в) частичного порядка;
г) линейного порядка;
д) эквивалентности.

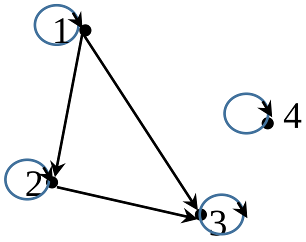
а)

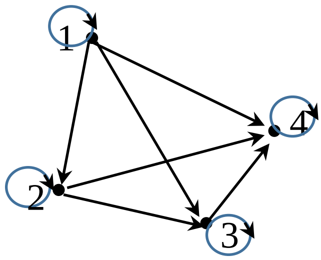


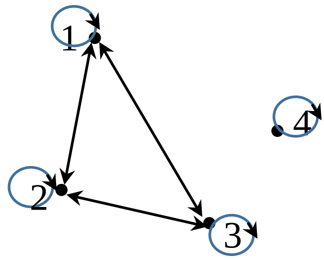
б)



в)



г) 

д) 

Теория алгоритмов.

Понятие алгоритма – одно из основных понятий математики.

Под алгоритмом понимается правило решения некоторого класса задач, разбитое на простые шаги, причём переход от одного шага к другому должен быть однозначно описан, и по окончании работы алгоритма должен быть достигнут некоторый результат. Итак, интуитивные требования к понятию алгоритма:

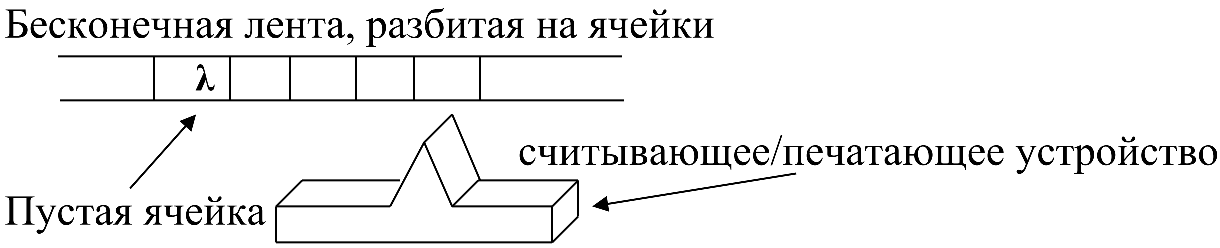
1. Алгоритм имеет дело с данными;
2. Алгоритм обладает памятью;
3. Элементарность шага. Алгоритм состоит из простых шагов, на каждом из которых выполняются некоторое простое действие;
4. Детерминированность. После выполнения каждого шага однозначно определено, что делать дальше;
5. Массовость. Алгоритм решает не одну простую задачу, а описывает решение некоторого класса задач;
6. Результативность. В конце работы алгоритма должен быть достигнут некоторый результат.

Всё, описанное выше, не может считаться строгим определением понятия алгоритма, поэтому ниже рассмотрим несколько математических конструкций, реализующих интуитивное понятие алгоритма.

Машина Тьюринга.

Представим бесконечную ленту, неограниченную ни вправо, ни влево, разбитую на ячейки. Пусть у нас имеется также некоторое устройство, способное перемещаться вдоль ленты вправо и влево, записывать в каждую ячейку по символу некоторого алфавита.

Устройство способно также считывать символы из каждой ячейки и перезаписывать информацию в эти ячейки. Устройство в каждый момент времени находится в некотором внутреннем состоянии, число которых конечно и своё для каждого устройства. Перемещение вдоль ленты, переход в другое состояние и печатаемый в ячейки символ зависит от того, в каком состоянии находилось устройство в предыдущий момент времени и от того, какой символ был считан из ячейки в предыдущий момент времени.



Договоримся *символ пустой ячейки* обозначать буквой λ.

Дадим строгое определение машины Тьюринга.

Машиной Тьюринга называется пятёрка $T = (A, S, \mu, \nu, \tau)$, где

$A = \{\lambda, a_1, a_2, ..., a_n\}$ - *алфавит*, то есть множество символов, которые могут записываться в ячейки ленты и считываться из этих ячеек. Каждый алфавит содержит символ пустой ячейки λ.

$S = \{s_0, s_1, s_2, ..., s_m\}$ - *множество внутренних состояний*, в которых может находиться машина Тьюринга, где s_0- *заключительное* состояние, s_1- *начальное* состояние;

$\mu : A \times S \to A$- *функция выхода*, показывающая, какой символ должна напечатать в обозреваемой ячейке машина Тьюринга в зависимости от того, какой символ прочитан на ленте в предыдущий момент времени и в каком состоянии находилась машина Тьюринга в предыдущий момент времени.

$\nu : A \times S \to S$- *функция перехода*, показывающая, в какое новое состояние должна перейти машина Тьюринга в зависимости от того, какой символ прочитан на ленте в предыдущий момент времени и в каком состоянии находилась машина Тьюринга в предыдущий момент времени.

$\tau: A \times S \to \{П, Л, Н\}$ - *функция управления*, показывающая, куда должно перемещаться печатающее устройство вдоль ленты: *П – вправо*, *Л – влево*, *Н –* оставаться *на месте*, в зависимости от того, какой символ прочитан на ленте в предыдущий момент времени и в каком состоянии находилась машина Тьюринга в предыдущий момент времени.

Считаем, что машина Тьюринга всегда начинает работу в состоянии s_1 и её считывающее устройство в это время зависает над первым слева непустым символом, записанным на ленте.

Если перед началом работы машины на ленте было записано слово α, а в результате работы произошёл переход в состояние s_0, то считается, что машина *закончила работу над словом* α, говорят, что машина Тьюринга *применима к слову* α, и итоговое слово, полученное на ленте, обозначается $T(\alpha)$. Если при работе над словом α машина Тьюринга не переходит в состояние s_0, то говорят, что машина Тьюринга *неприменима к слову* α.

Для описания процесса работы машины Тьюринга над словом будем использовать последовательность так называемых конфигураций.

Конфигурацией машины Тьюринга будем называть запись вида $\alpha \underset{s_p}{a_i} \beta$, где α – начальный отрезок слова, β - заключительный отрезок слова, над которым происходит работа машины, a_i - символ, обозреваемый в данный момент времени, а s_p - состояние, в котором находится машина в настоящий момент времени.

Сводная таблица трёх функций – выхода, перехода и управления, называется *программой* машины Тьюринга.

Если в данный момент времени машина находится в состоянии s_p, a_i- символ, обозреваемый в данный момент времени, $\mu(a_i, s_p) = a_m$, $\nu(a_i, s_p) = s_r$, $\tau(a_i, s_p) = D$, $D \in \{П, Л, Н\}$, то запись

$a_i s_p \to a_m D s_r$ называется *команδой* машины Тьюринга.

Пример. Рассмотреть работу машины Тьюринга, заданную программой, записанной в таблице, над словом $\alpha = abba$ и записать её в виде последовательности конфигураций.

S / A	S_1	S_2
λ	----	bЛS_0
a	bПS_1	aПS_2
b	aПS_2	aНS_0

что пишем   в какое состояние переходим

куда двигаемся (вправо, влево или на месте)

Начальная конфигурация: $\underset{s_1}{a}bba$. По таблице видим, что, если машина находится в состоянии s_1 и считывается символ a, то нужно печатать символ b, двигаться вдоль ленты вправо и оставаться в состояние s_1. Получили новую конфигурацию $b\underset{s_1}{b}ba$. Продолжаем изображать последовательность конфигураций, выпишем все конфигурации: $\underset{s_1}{a}bba,\ b\underset{s_1}{b}ba,\ ba\underset{s_2}{b}a,\ baa\underset{s_0}{a}$.

Получили, что машина Тьюринга применима к слову $abba$ и $T(abba) = baaa$.

В дальнейшем повторение n раз подряд символа a будем для краткости заменять на a^n. Тогда можем записать: $T(ab^2a) = ba^3$.