Цель лекции: дать основные понятия алгоритмов теории графов.

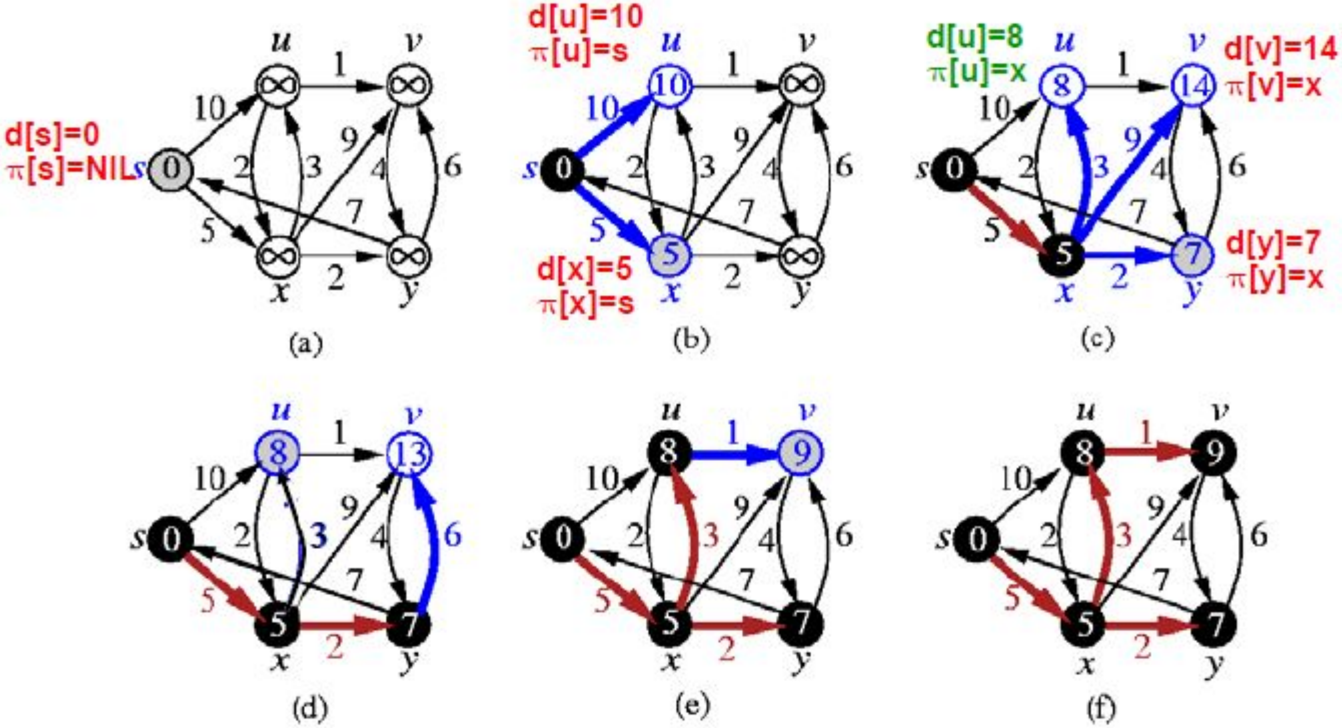
Алгоритм построения кратчайшего пути

Алгоритм Дейкстры

1. Заполнить массив *up* [1..*n*] нулями.
2. Каждой вершине *i* приписать в качестве ключа *dist* [*i*] – максимально возможное число (оно должно быть больше, чем длина наибольшего из кратчайших путей в графе; в процессе вычислений это число будет уменьшаться и в итоге заменится на длину кратчайшего пути из вершины *s* в вершину *i*).
3. Организовать приоритетную очередь из вершин графа, взяв в качестве ключей величины *dist* [*i*], *i*= 1, 2, …, *n*.
4. Заменить ключ вершины *s* на 0.
5. Пока очередь не пуста, выполнять операции 6, 7.
6. Выбрать (с удалением) из приоритетной очереди элемент r_0 с минимальным ключом.
7. Для каждой вершины *r*, смежной с r_0, выполнить операции 8, 9.
8. Вычислить величину $delta = dist[r] - (dist[r_0] + L(r_0, r))$.
9. Если $delta > 0$, то уменьшить ключ *dist* [*r*] элемента *r* на величину *delta* и заменить старое значение величины *up* [*r*] на r_0.

Алгоритм построения кратчайшего пути

Алгоритм Дейкстра



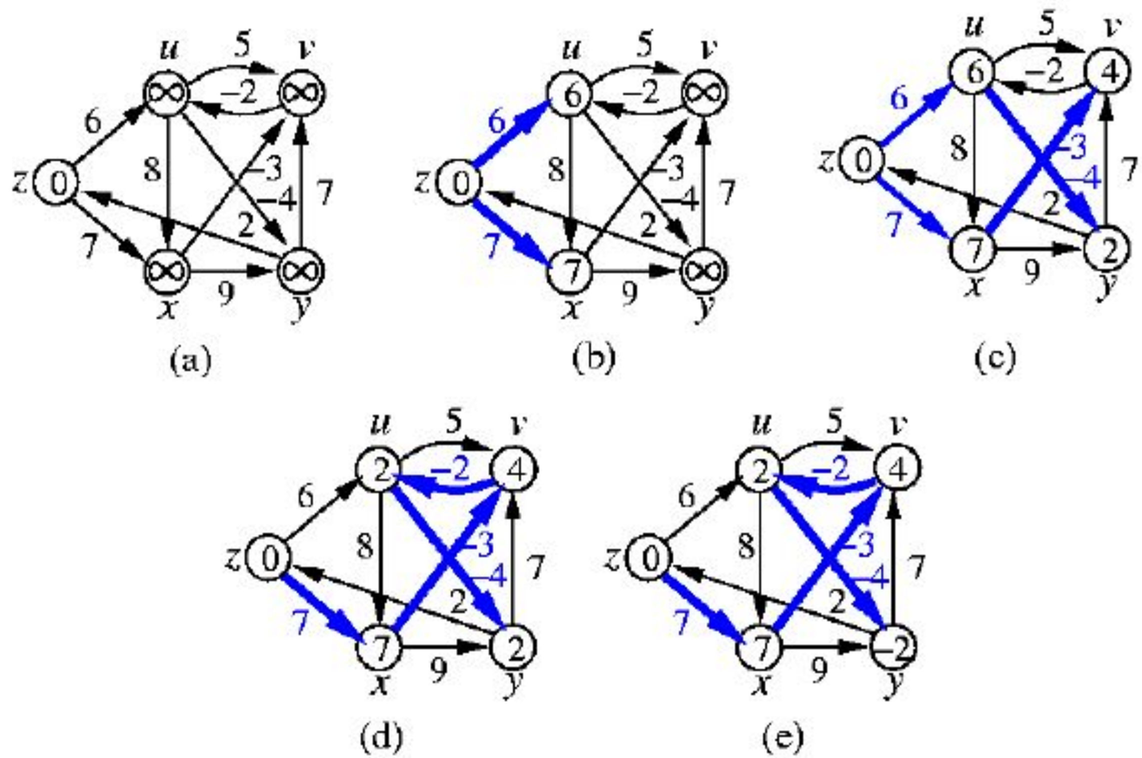
Алгоритм построения кратчайшего пути

Алгоритм Беллмана-Форда

```
Initialize-Single-Source(G, s)
for  i = 1 to |G.V| - 1
      for each edge (u, v) ∈ G.E
            Relax(u, v, w)
for each edge (u, v) ∈ G.E
      if v.d > u.d + w(u, v)
            return FALSE
return TRUE
```

Алгоритм построения кратчайшего пути

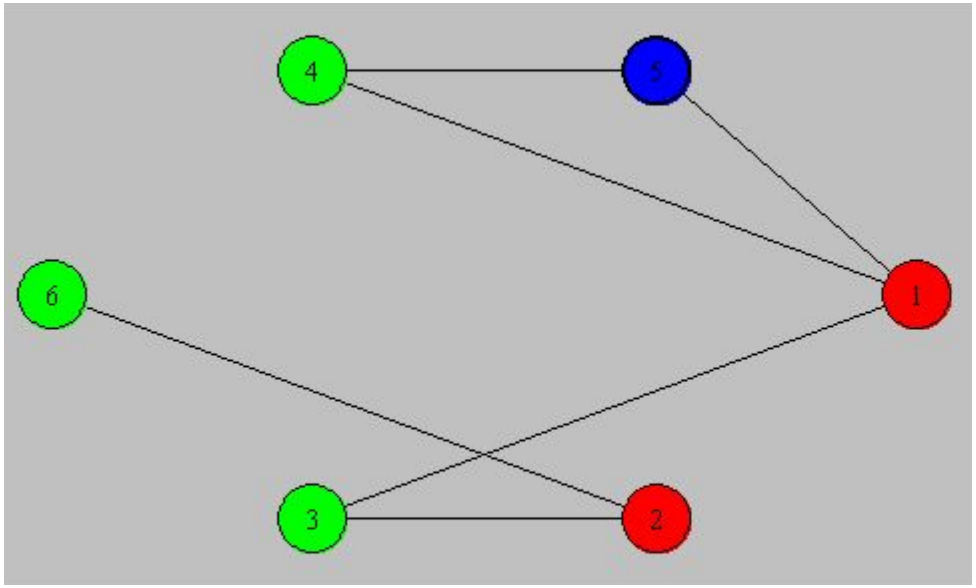
Алгоритм Беллмана-Форда



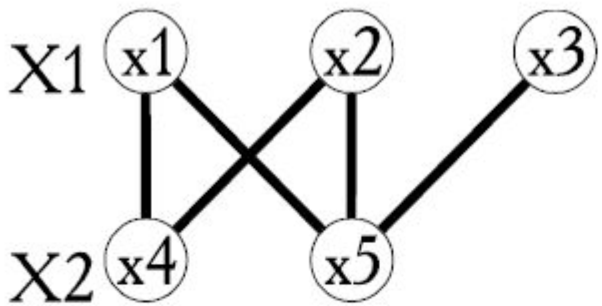
Хроматическое число графа

Раскраской вершин графа называется разбиение множества вершин X на l непересекающихся классов (подмножеств):

$$X1, X2, \ldots, X_l; X = \bigcup_{i=1}^{l} X_i; X_i \bigcap X_j = \varnothing; i, j \in I = \{1, 2, \ldots, l\}, \quad (3.6)$$



Двухдольный граф



Алгоритм раскраски

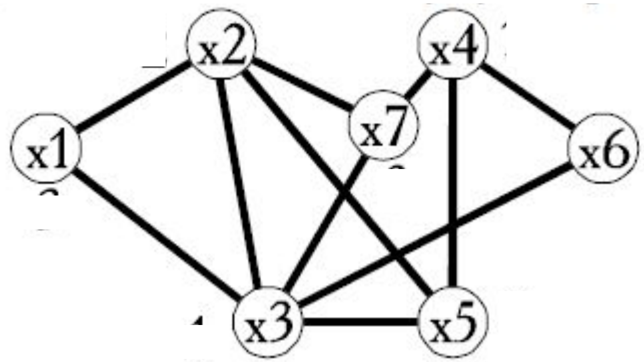
Алгоритм неявного перебора при раскраске графа

Предположим, что множество вершин как-то упорядочено и x_i — i-я вершина этого множества. Тогда первоначальная допустимая раскраска может быть получена так:

1. Окрасить x_i в цвет 1.
2. Каждую из оставшихся вершин окрашивать последовательно: вершина x_i окрашивается в цвет с наименьшим возможным «номером» (т. е. выбираемый цвет должен быть первым в данном упорядочении цветом, не использованным при окраске какой-либо вершины, смежной x_i).

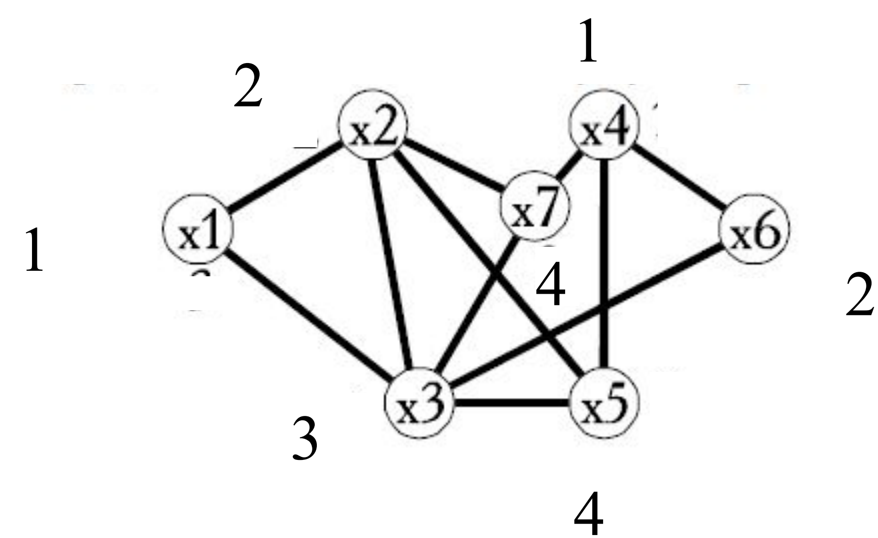
Алгоритм раскраски

Алгоритм неявного перебора при раскраске графа



Алгоритм раскраски

Алгоритм неявного перебора при раскраске графа



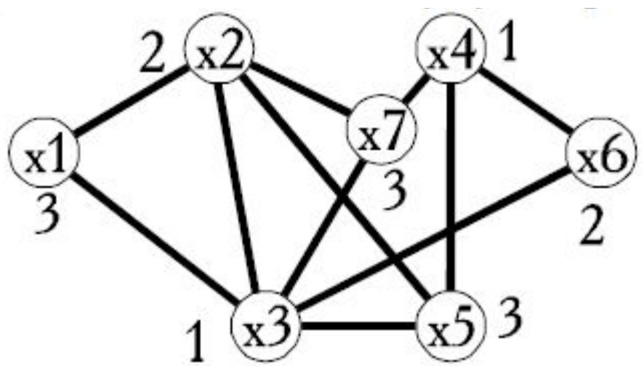
Алгоритм раскраски

Последовательный алгоритм раскраски графа

Шаг 1. Составить упорядоченный в порядке убывания степеней вершин список.
Шаг 2. Первая вершина окрашивается в цвет 1 и удаляется из списка.
Шаг 3.Просматривая список, в текущий цвет раскрашиваются и удаляются из списка все вершины, не смежные с выбранной и между собой. Номер цвета увеличивается на 1.
Шаг 4. Далее выбирается первая вершина из списка, она окрашивается в текущий цвет и удаляется.
Шаг 5. Процесс продолжается, пока не будут окрашены все вершины.

Алгоритм раскраски

Последовательный алгоритм раскраски графа. Пример.



x(i)	x3	x2	x4	x5	x7	x1	x6
ρ(x(i))	5	4	3	3	3	2	2

x(i)	x2	x5	x7	x1	x6
ρ(x(i))	4	3	3	2	2

x(i)	x5	x7	x1
ρ(x(i))	3	3	2

Алгоритм раскраски

Применение

- Задачи расписания
- Распределение ресурсов
- Распределение регистров в микропроцессорах
- Распределение частот для мобильной связи
- Задачи ЭМС
- и др.

Основные выводы:

1. Изучены основные понятия алгоритмов теории графов;
2. Рассмотрено применение теории при решении задач конструкторско-технологической информатики;