Алгоритмы на графах

Модуль 2. Кратчайшие расстояния.

Лекция 6.
Алгоритм Краскала.

Адигеев Михаил Георгиевич

2024

План лекции

1. Алгоритм Краскала
   - ✓ Алгоритм
   - ✓ Реализация на основе Union-Find

Минимальное остовное дерево

Жадный алгоритм — однопроходный итерационный алгоритм. Строит решение, добавляя на каждом шаге к текущему частичному решению новый элемент. Добавляемый элемент выбирается на основе локального оптимума («наилучший на текущем шаге»).

Жадные алгоритмы:

- ✓ Дают точное решение для задач на матроидах (с аддитивной целевой функцией)
- ✓ Для некоторых задач дают приближённое решение с гарантированной (не обязательно константной) оценкой приближения
- ✓ В общем случае используются как эвристики

Минимальное остовное дерево

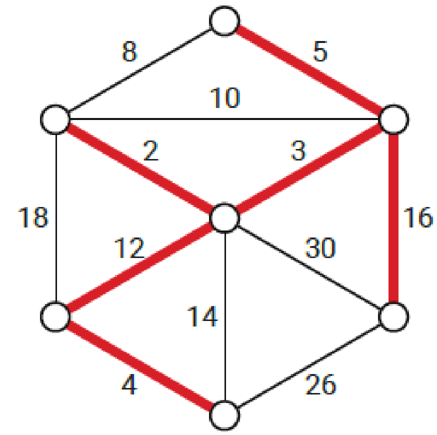
Жадная стратегия применительно к построению минимального остовного дерева:

1. Начинаем с пустого подграфа $G'(V', E')$.
2. Итерационно добавляем к текущему подграфу G' новое ребро (и вершину?), при условии, что подграф остаётся ациклическим. Такое ребро будем называть ***безопасным***.
3. Завершаем, когда $|E'| = n - 1 = |V| - 1$.

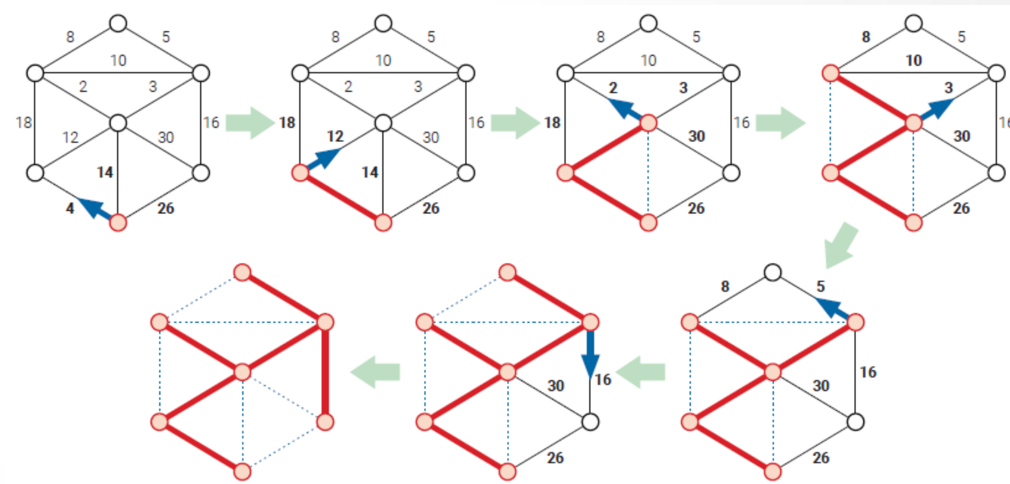
Эта стратегия по-разному использована в двух классических алгоритмах:

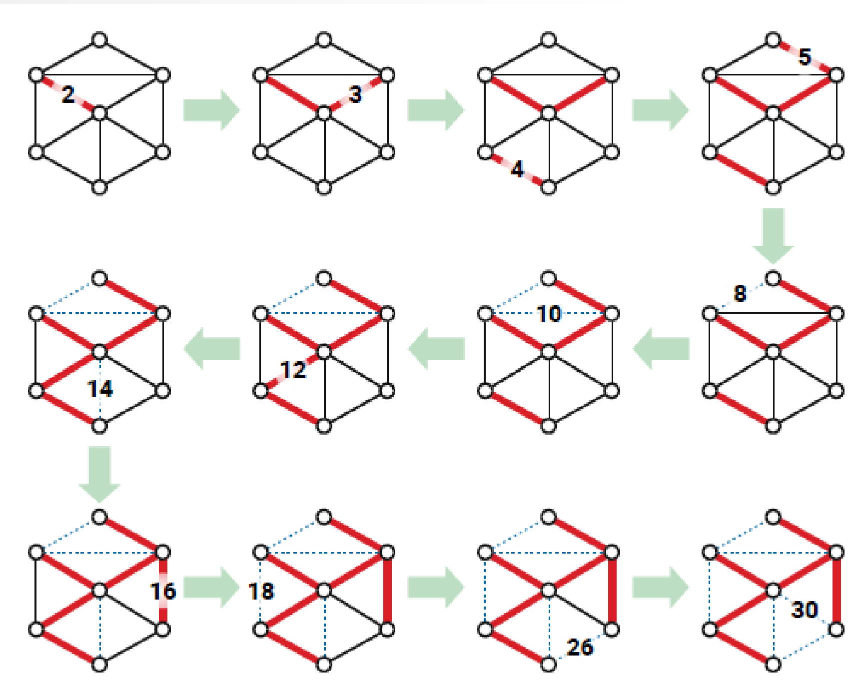
- ✓ Алгоритм Краскала: строим остовный лес, добавляя новые рёбра, пока не получим одно дерево.
- ✓ Алгоритм Прима: строим дерево, добавляя новые рёбра и вершины, пока не получим остовное дерево.

Минимальное остовное дерево



Prim's algorithm





Kruskal's algorithm

http://jeffe.cs.illinois.edu/teaching/algorithms/

Алгоритм Краскала

Алгоритм Краскала

Вход: связный неориентированный граф $G(V, E)$, $|V| = n, |E| = m$.

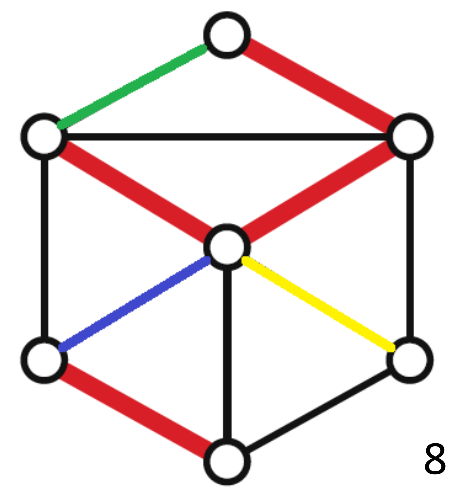
Выход: минимальное остовное дерево $G'(V, E')$.

Решение будем строить итерационно. На каждом шаге текущее решение – остовный лес $G'(V, E')$. Т.к. подграф G' всегда содержит все вершины, достаточно отслеживать множество рёбер E'.

Алгоритм Краскала

1. $E' \coloneqq \emptyset$
2. Отсортировать множество всех рёбер графа по увеличению весов.
3. For each $e \in E$ в порядке увеличения весов:
   - Если ребро e безопасно, то $E' \coloneqq E' \cup \{e\}$

Как проверять безопасность ребра?



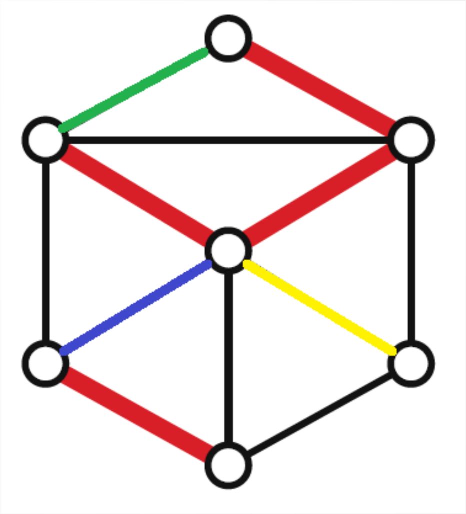
Алгоритм Краскала

Как проверять безопасность ребра?

1) Условно добавить ребро к E' и проверить наличие цикла на графе. Сложность проверки одного ребра: $O(n + m') = O(n)$. Общая сложность шага (3) и алгоритма в целом: $O(m \cdot n) = O(n^3)$.

Алгоритм Краскала

2) Вести список компонент связности леса G'. Ребро $e = (u, v)$ безопасно тогда и только тогда, когда u и v лежат в разных компонентах.



Ключевые операции:

✓ Получение номера компоненты для заданной вершины.

✓ Объединение двух компонент.

Алгоритм Краскала

При использовании специализированной структуры для хранения непересекающихся множеств Union_Find:

- ✓ Получение номера компоненты для заданной вершины. $O(\log n)$
- ✓ Объединение двух компонент. $O(\log n)$

Алгоритм Краскала

1. $E' \coloneqq \emptyset$
2. Отсортировать множество всех рёбер графа по увеличению весов. $O(m \log m)$
3. For each $e \in E$ в порядке увеличения весов:
   - Если ребро e безопасно, то $E' \coloneqq E' \cup \{e\}$

   $O(m \log n)$

Так как $m = O(n^2)$, итоговую сложность алгоритма можно записать как $O(m \cdot \log n)$.

Алгоритм Краскала

Осталось доказать, что алгоритм Краскала строит минимальное остовное дерево.

Для этого вспомним теорему о деревьях и докажем новую теорему – «Цепной критерий оптимальности остовного дерева».

Деревья

<u>Теорема о деревьях</u>.

Для конечного графа $G(V,E)$ следующие утверждения эквивалентны:

1) G — дерево.

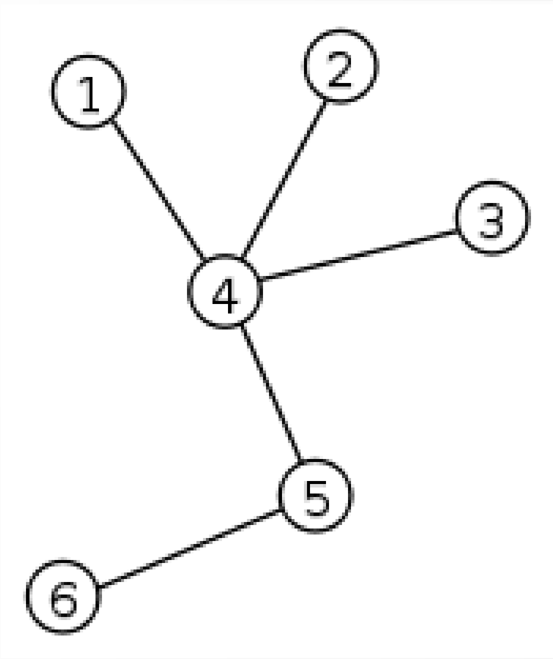
2) G — не содержит циклов и $|E| = |V| - 1$.

3) G — связен и $|E| = |V| - 1$.

4) G — связен и каждая дуга является мостом.

5) Любые две вершины можно соединить единственной простой цепью.

6) G не содержит циклов, и добавление к нему любой новой дуги приводит к образованию единственного простого цикла.



Цепной критерий оптимальности

Определение.

Пусть G' - остовное дерево графа G, ребро (u, v) принадлежит дереву G'. При удалении (u, v) из G' дерево G' распадается на 2 компоненты (п.4 из теоремы о деревьях). Через $\delta(G', (u, v))$ обозначим полученный ***разрез***, то есть множество рёбер, у которых концы лежат в разных компонентах полученного леса.

Цепной критерий оптимальности

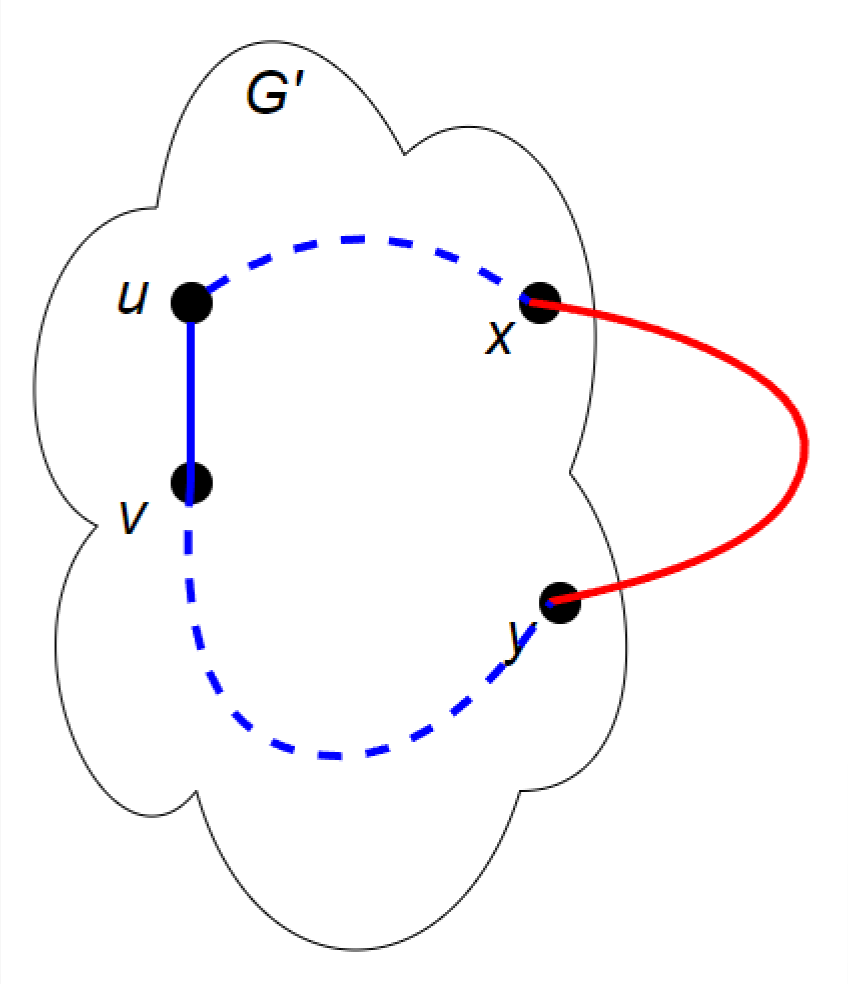
<u>**Теорема**</u> (Цепной критерий оптимальности остовного дерева).

Остовное дерево $G'(V', E')$ минимально $\Leftrightarrow$ для любого *недревесного* ребра $(x, y) \in E \backslash E'$ и любого *древесного* ребра $(u, v) \in E'$, лежащего на цепи, соединяющей в G' вершины x и y,выполняется: $w(u, v) \leq w(x, y)$.

Доказательство.

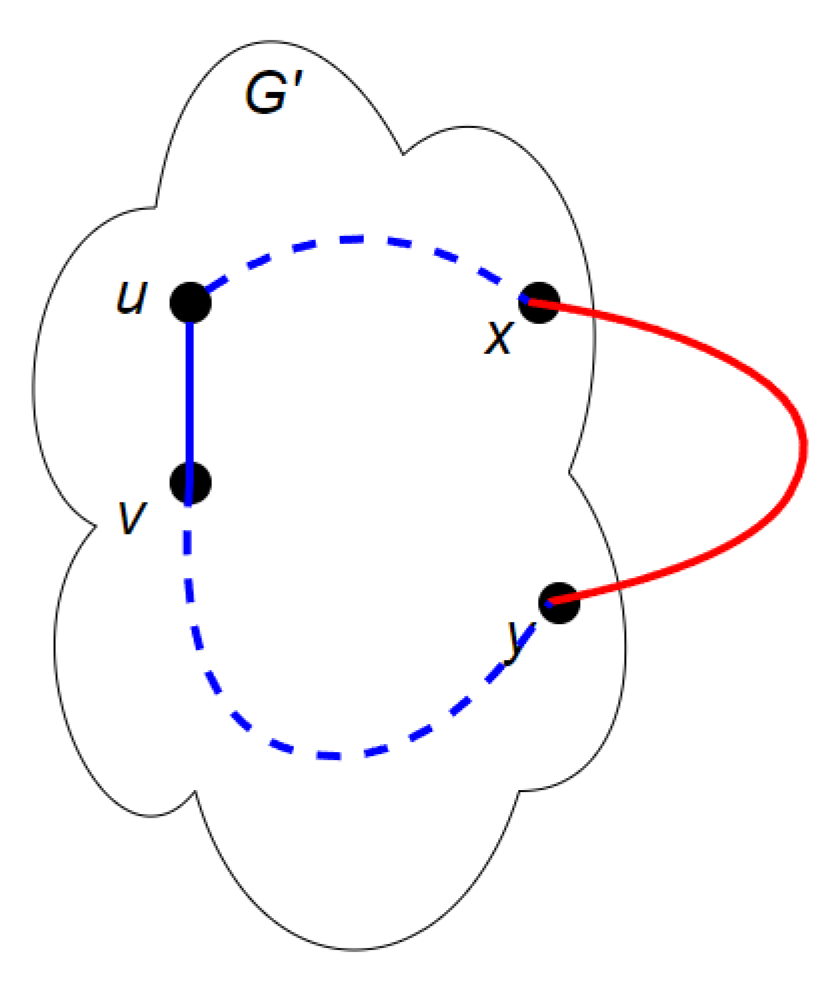
Цепной критерий оптимальности

Доказательство.



$\Rightarrow$ Пусть G' - минимальное остовное дерево. Граф $H = (G' \setminus \{(u, v)\}) \cup \{(x, y)\}$ – тоже остовное дерево (применяем пп. 6 и 5 из теоремы о деревьях. Поэтому $w(G') \le w(H)$, и, следовательно, $w(u, v) \le w(x, y)$.

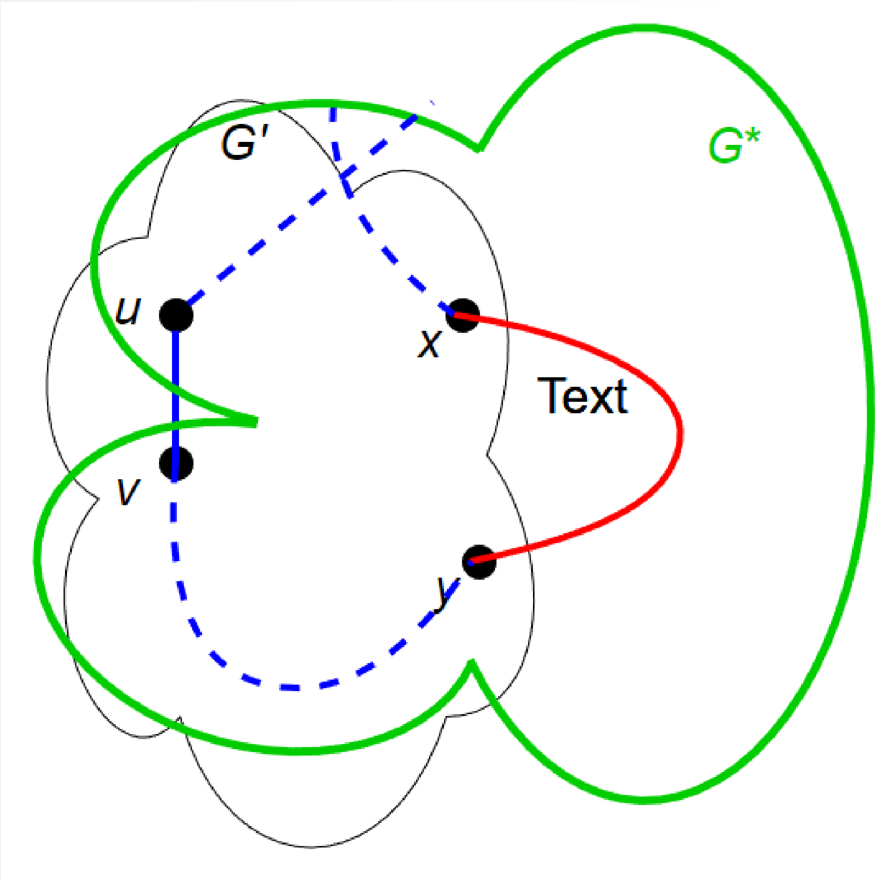
Цепной критерий оптимальности



$\Leftarrow$ Пусть для G' выполняется условие теоремы, но G' - не минимальное остовное дерево.

Пусть G^* - то из минимальных остовных деревьев, которое имеет максимально возможное количество общих рёбер с G'. Выберем в качестве (x, y) такое ребро, которое принадлежит G^* и не принадлежит G' ($(x, y) \in G^* \backslash G'$).

Цепной критерий оптимальности



В качестве ребра (u, v) выберем ребро, на цепи, соединяющей в G' вершины x и y, и принадлежащее *разрезу* $\delta(G^*, (x, y))$.

Построим граф $H = (G^* \setminus \{(x, y)\} \cup \{(u, v)\}$. Это остовное дерево и $w(H) \leq w(G^*)$. Поэтому *H* – тоже минимальное остовное дерево. Но *H* содержит больше общих рёбер с G' , чем G^* . Противоречие с выбором G^*.

Алгоритм Краскала

Теорема. Алгоритм Краскала строит минимальное остовное дерево.

Доказательство.

Пусть $G'(V, E')$ - результат работы алгоритма Краскала. По построению алгоритма, G' - ациклический остовный подграф. Докажем, что он связен и минимален.

1. G' связен, т.е. является деревом. Предположим, что это не так. Тогда существует ребро e, такое что $G' \cup \{e\}$ не содержит цикла. Тогда ребро e безопасно и должно было быть добавлено на одной из итераций.

Алгоритм Краскала

2. Покажем, что G' - <u>минимальное</u> остовное дерево.

Покажем, что G' удовлетворяет цепному критерию оптимальности. Пусть (x, y) - произвольное недревесное ребро. Тогда при рассмотрении этого ребра оно оказалось небезопасным $\Rightarrow$ все рёбра на цепи, соединяющей в G' вершины x и y, уже были добавлены ранее $\Rightarrow$ их веса меньше, чем вес ребра (x, y).

Теорема доказана.