

Алгоритмы на графах

Модуль 2. Кратчайшие расстояния.

Лекция 8.

Кратчайшие пути (часть 1).

Адигеев Михаил Георгиевич

2022

План лекции

1. Задача о кратчайшем пути
 - ✓ Почему не подходят поиск в ширину и топологическая сортировка?
2. Алгоритм Дейкстры
 - ✓ Общая схема алгоритма
 - ✓ Обоснование корректности
 - ✓ Эффективная реализация
 - ✓ Как работает с отрицательными весами дуг?
3. Задача о кратчайшем пути при отрицательных весах.
4. Алгоритм Беллмана-Форда.

Задачи о кратчайшем пути

Задача о кратчайшем пути

Дан взвешенный граф $G(V, E)$, $w: E \rightarrow R$.

Вес пути определим как сумму весов входящих в него дуг.

Кратчайший путь = путь минимального веса.

Возможны три формулировки задачи:

- 1) Заданы вершины $s, t \in V$. Найти кратчайший путь из s в t .
- 2) Задана вершина $s \in V$. Найти кратчайшие пути из s во все остальные вершины графа.
- 3) Найти кратчайшие пути для всех пар вершин на графе.

Сейчас рассматриваем варианты (1) и (2).

Задачи о кратчайшем пути

Мы уже рассматривали 2 алгоритма для поиска кратчайших путей

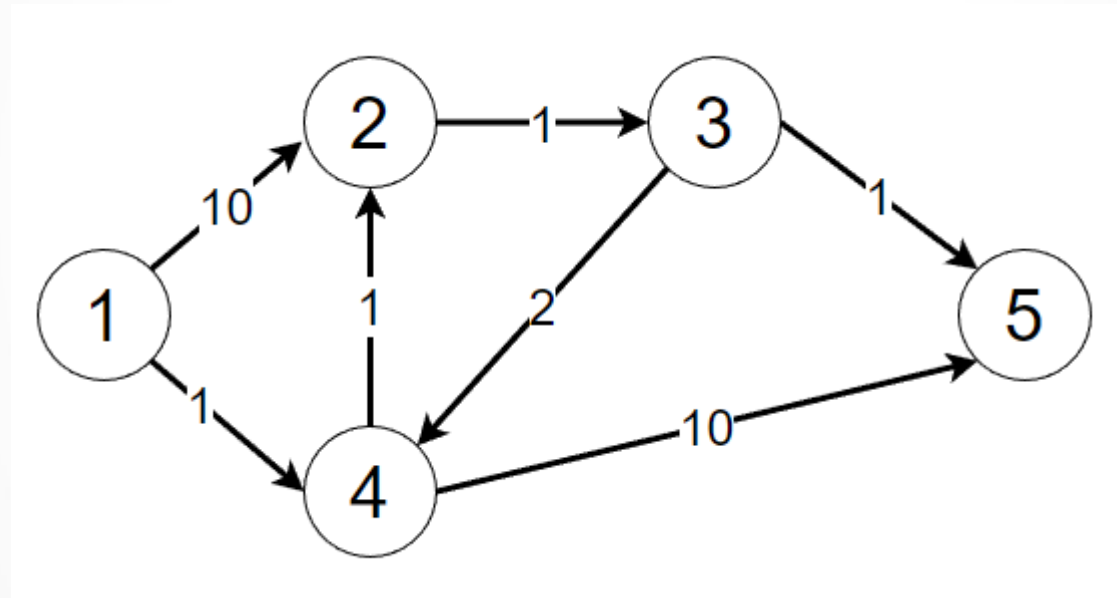
- Алгоритм на основе поиска в ширину.
- Алгоритм на основе топологической сортировки.

Почему их не достаточно?

Задачи о кратчайшем пути

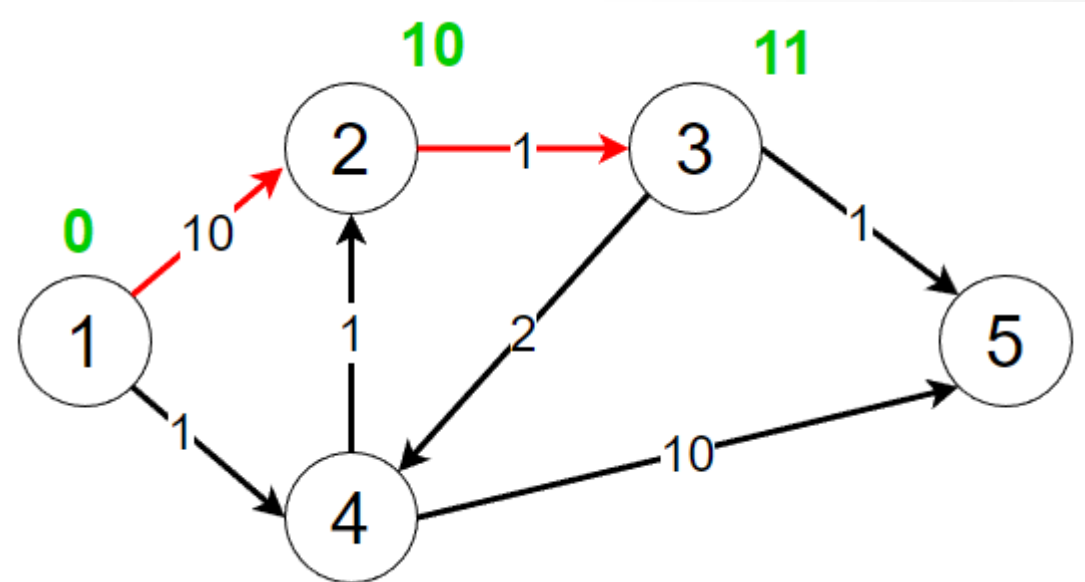
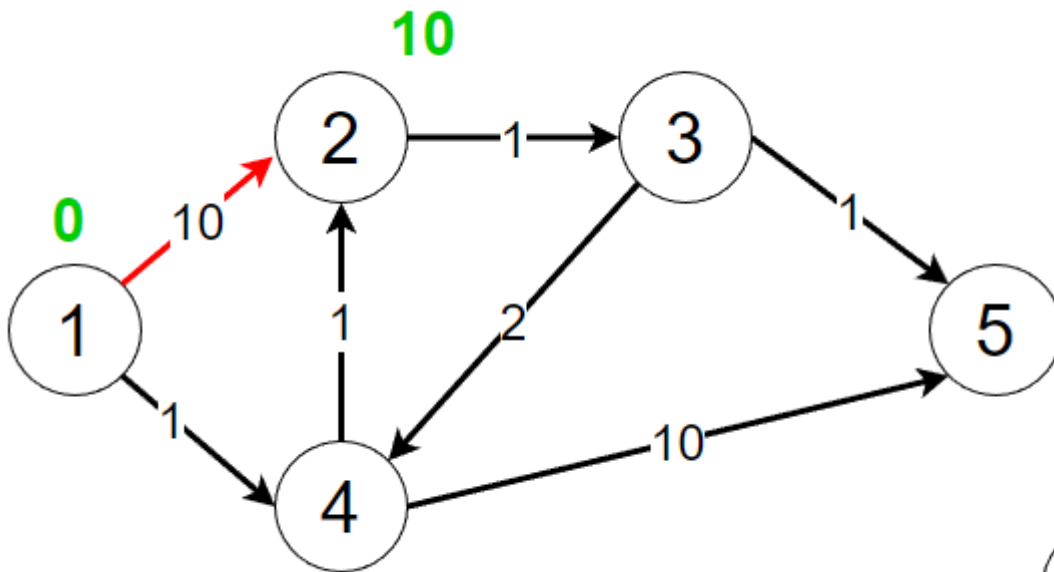
Топологическая сортировка

Рассмотрим случай, когда на графе есть контур.



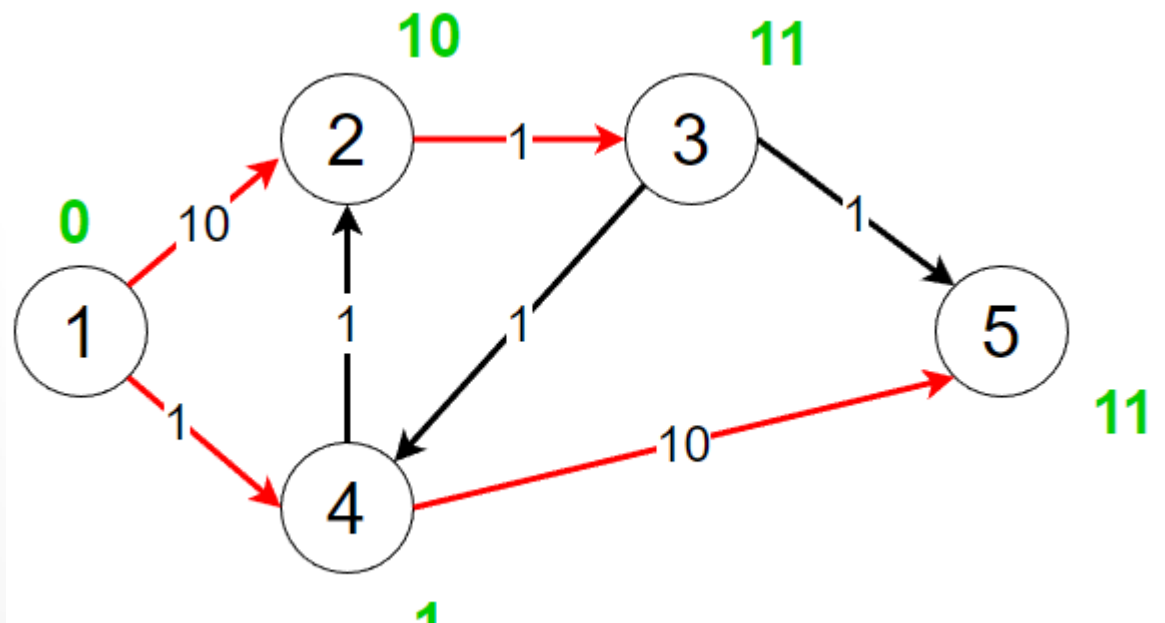
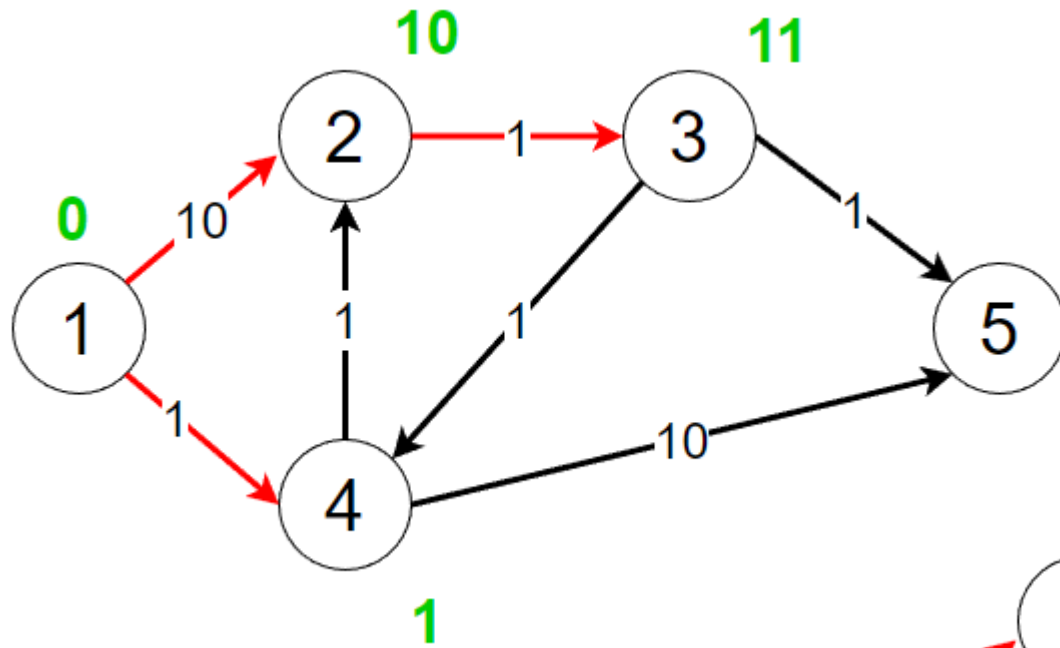
Задачи о кратчайшем пути

Топологическая сортировка



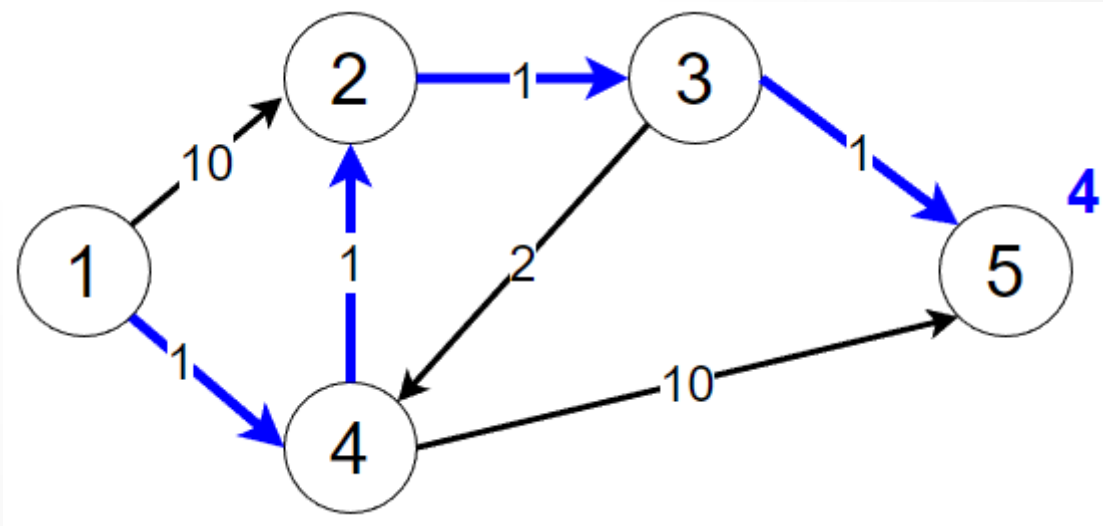
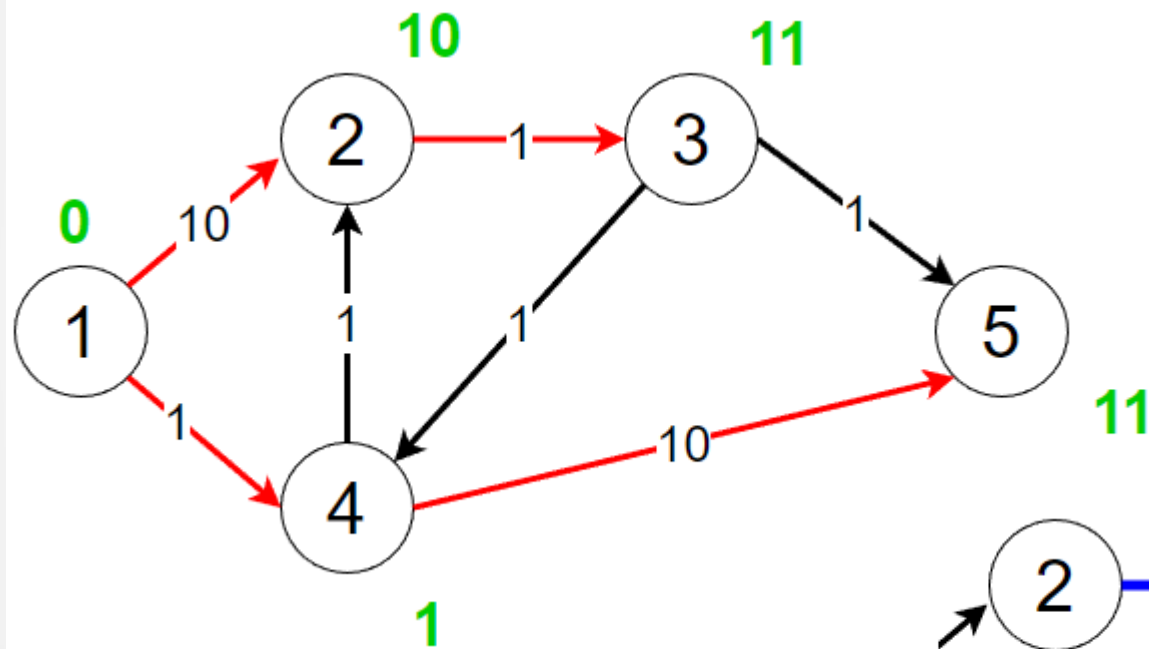
Задачи о кратчайшем пути

Топологическая сортировка



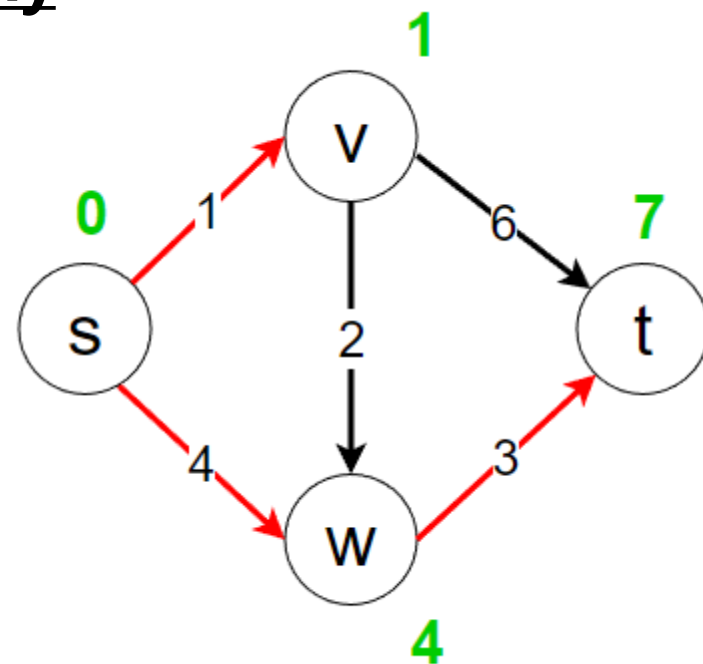
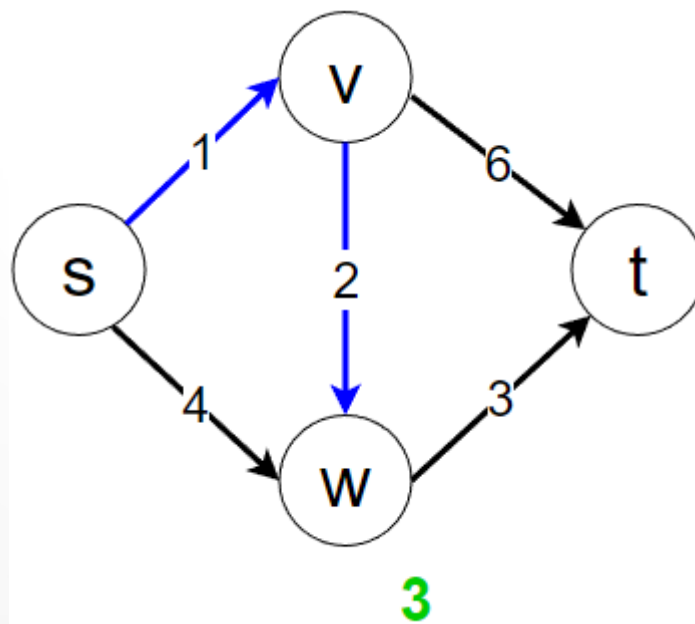
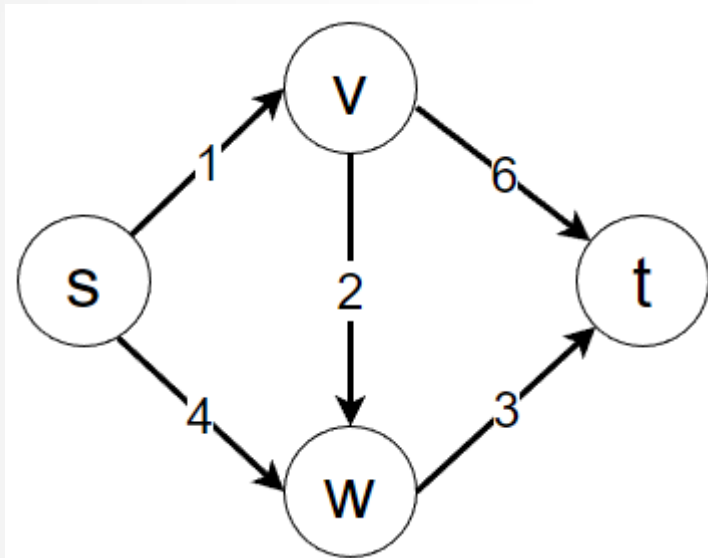
Задачи о кратчайшем пути

Топологическая сортировка



Задачи о кратчайшем пути

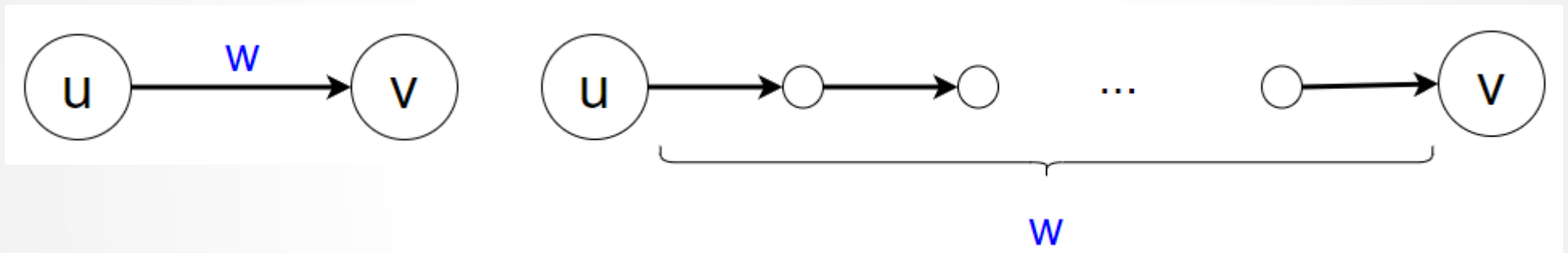
Поиск в ширину



Задачи о кратчайшем пути

Поиск в ширину

Но можно изменить граф – заменить ребро веса w путём из w дуг.



В этом случае алгоритм отработает корректно.

Но сложность такого алгоритма: $O(n' + m') = O(\sum w_i)$ – эта величина при больших значениях весов оказывается слишком большой.

Алгоритм Дейкстры

Алгоритм Дейкстры

Дан взвешенный ориентированный граф $G(V, E)$, $w: E \rightarrow R_+$.

Вес пути определим как сумму весов входящих в него дуг.

Кратчайший путь = путь минимального веса.

Задана вершина $s \in V$. Найти кратчайшие пути из s во все остальные вершины графа.

Алгоритм Дейкстры по структуре похож на обход в ширину и особенно на алгоритм Прима:

- 1) Создаём подмножество вершин V' , инициализируем вершиной s .
- 2) Итерационно добавляем в V' *самую лучшую* вершину.
- 3) Завершаем алгоритм, когда $V' = V$.

Алгоритм Дейкстры

Алгоритм Дейкстры

$V' := \{s\}$

For each $v \in V$:

$d[v] := +\infty$;

$p[v] := \text{NULL}$;

$d[s] := 0$;

Пока существует дуга $(u, v) : u \in V', v \notin V'$:

 Выбрать дугу $(u^*, v^*) : d[u^*] + w[u^*, v^*]$ минимально;

$V' := V' \cup \{v^*\}$;

$d[v^*] := d[u^*] + w[u^*, v^*]$;

$p[v^*] := u^*$;

 Update_C&P(v^*);

Update C&P(v)

For each $(v, u) \in E$:

 if $u \in V \setminus V'$ & $d[u] > d[v] + w[v, u] : d[u] := d[v] + w[v, u]$;

Алгоритм Дейкстры

Теорема Алгоритм Дейкстры корректно находит расстояния от s до вершин графа, при условии, что веса всех дуг неотрицательны.

Доказательство

Доказательство проведём с помощью полной математической индукции.

Параметр индукции: порядковый номер k , в порядке добавления вершин в V' .

Утверждение: для всех $v \in V'$ величина $d[v]$ равна расстоянию от s до v .

1. Базис индукции: $k = 1$.

Это значит, что $v = s$. Для s выполнено: $d[s] = 0 =$ расстоянию от s до s .

2. Индуктивное предположение: для вершин с номерами $\leq k$ выполнено доказываемое утверждение.

Алгоритм Дейкстры

3. Индуктивный переход: выберем вершину v^* , добавленную $(k + 1)$ -й.

Пусть (u^*, v^*) – дуга, выбранная в алгоритме перед добавлением v^* .

Вершина u^* добавлена в V' раньше. По предположению индукции, $d[u^*]$ = расстоянию от s до u^* . Тогда $d[v^*] = d[u^*] + w[u^*, v^*]$ – вес пути из s в v^* через u^* . Предположим, что есть более короткий путь (путь меньшего веса) π' .

Путь π' начинается в вершине $s \in V'$ и оканчивается в $v^* \notin V'$. Пусть (u', v') – первая на π' дуга, такая, что $u' \in V', v' \notin V'$.

Алгоритм Дейкстры

Реально алгоритмы Дейкстры строят дерево кратчайших путей. Дерево определяется ссылками $p[v]$.

Какова сложность алгоритма? Она зависит от реализации операций

Выбрать дугу $(u^*, v^*) : d[u^*] + w[u^*, v^*]$ минимально;

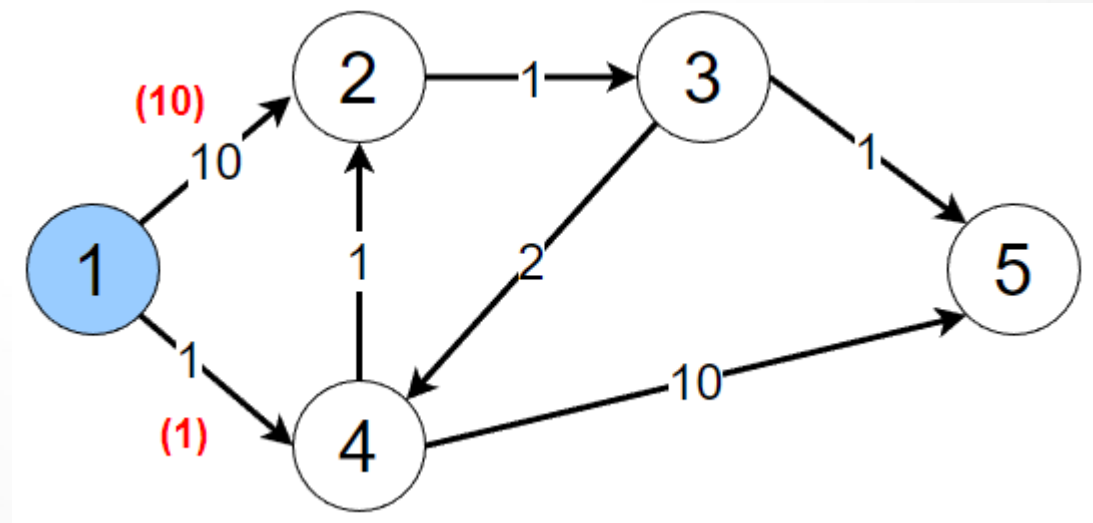
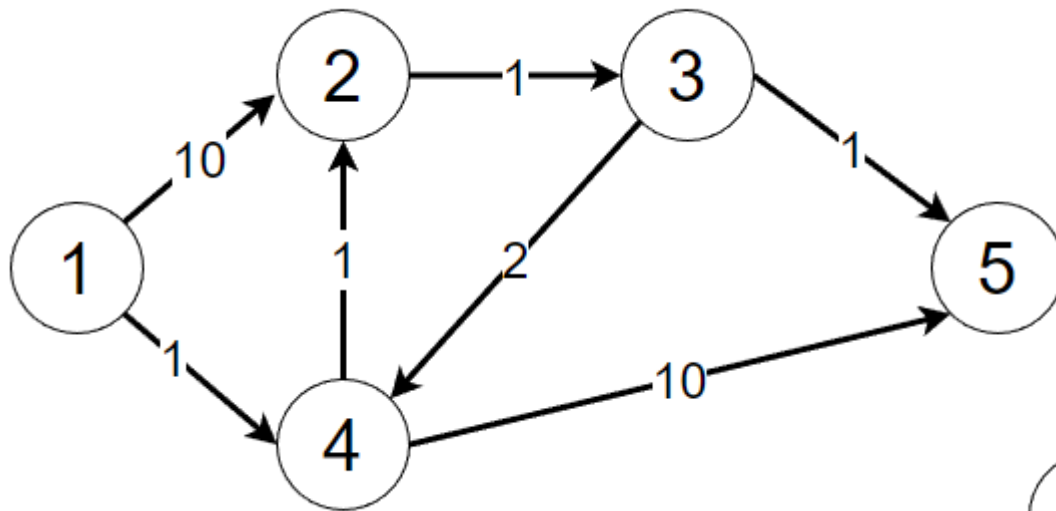
Наивная реализация: на каждой итерации ($(n - 1)$ итераций) перебираем все дуги и для каждой проверяем условие $(u, v) : u \in V', v \notin V'$ и выбираем оптимальную дугу. Сложность: $O(nt)$.

Оптимальная реализация: дуги хранить в очереди с приоритетами.

Сложность: $O((n + t) \log n)$.

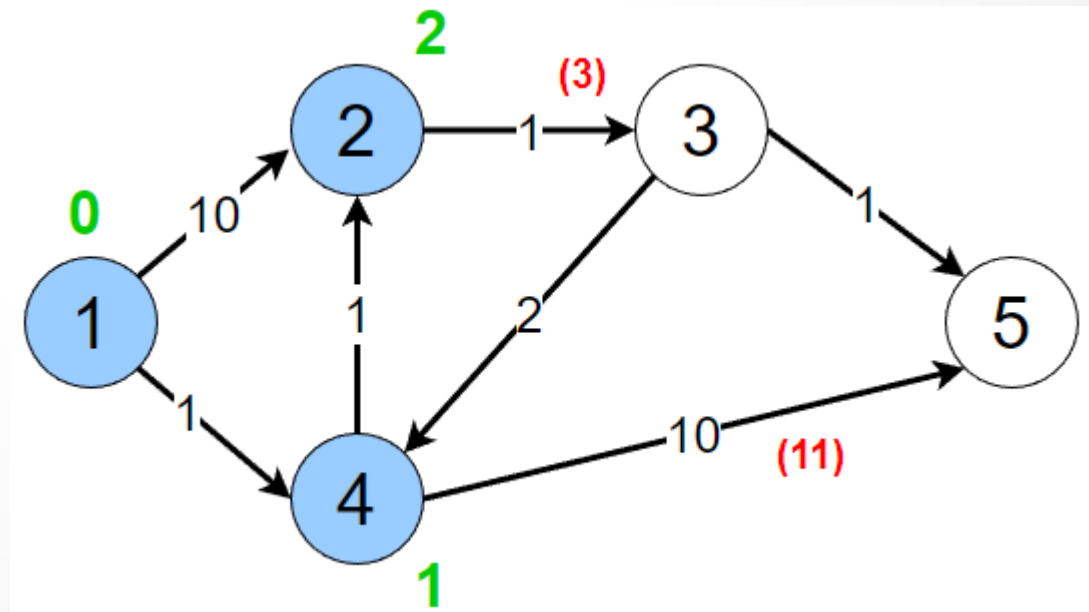
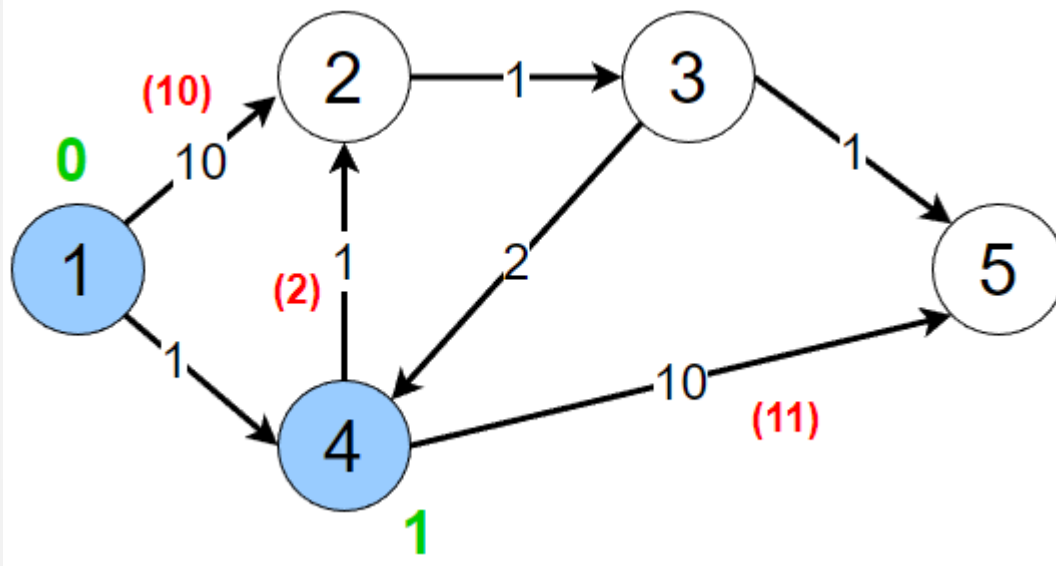
Алгоритм Дейкстры

Посмотрим, как алгоритм Дейкстры обрабатывает граф с контуром.



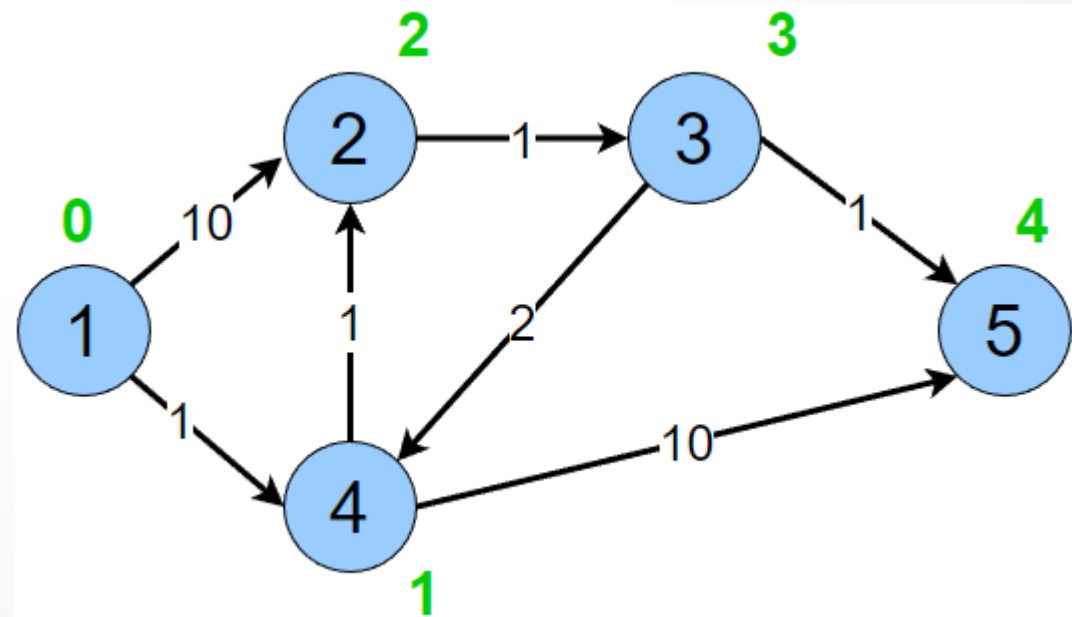
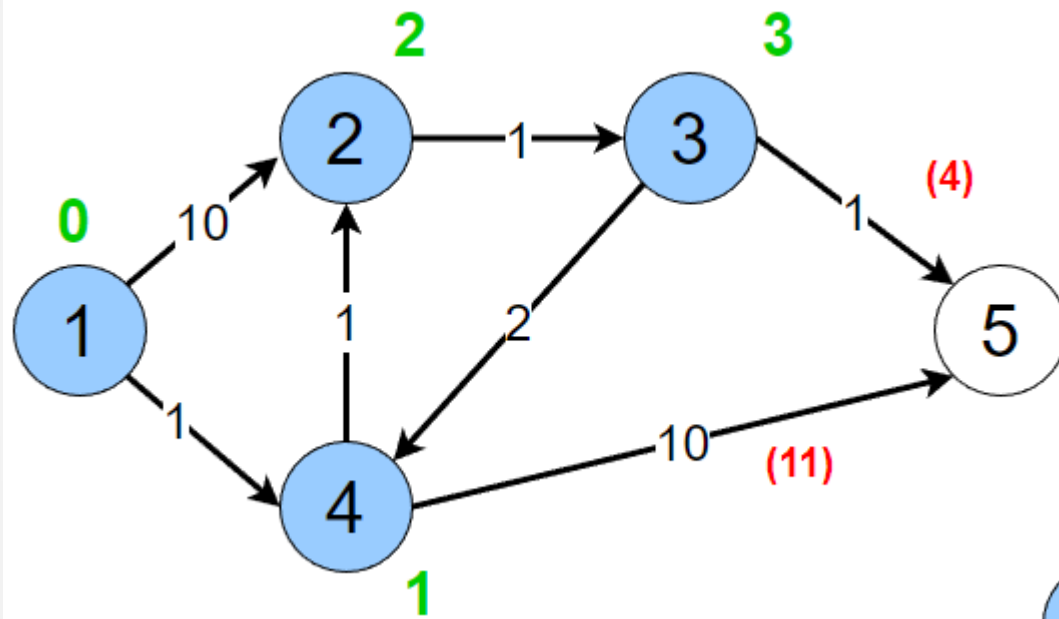
Алгоритм Дейкстры

Посмотрим, как алгоритм Дейкстры обрабатывает граф с контуром.



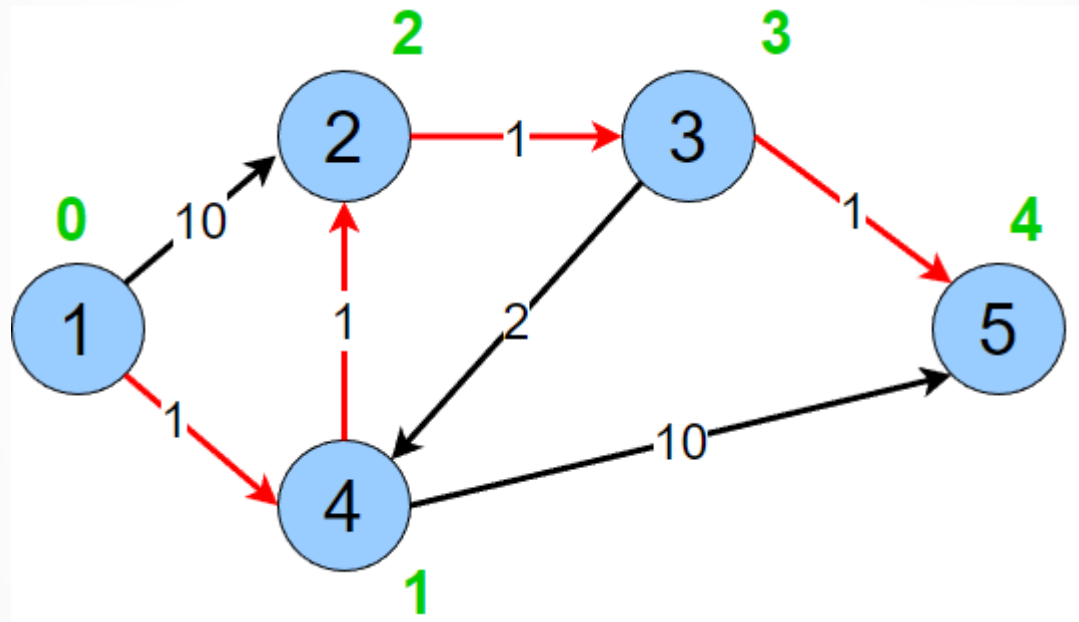
Алгоритм Дейкстры

Посмотрим, как алгоритм Дейкстры обрабатывает граф с контуром.



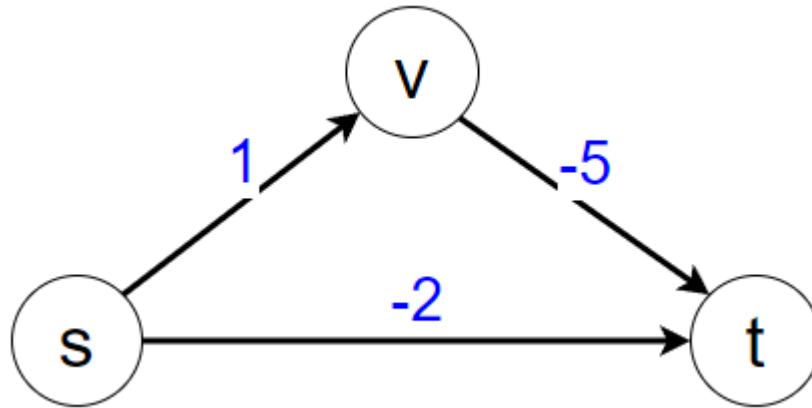
Алгоритм Дейкстры

Посмотрим, как алгоритм Дейкстры обрабатывает граф с контуром.



Алгоритм Дейкстры

Как алгоритм Дейкстры работает с графами при наличии отрицательных весов дуг?



Инициализация: $V' = \{s\}$, $d[s] = 0$

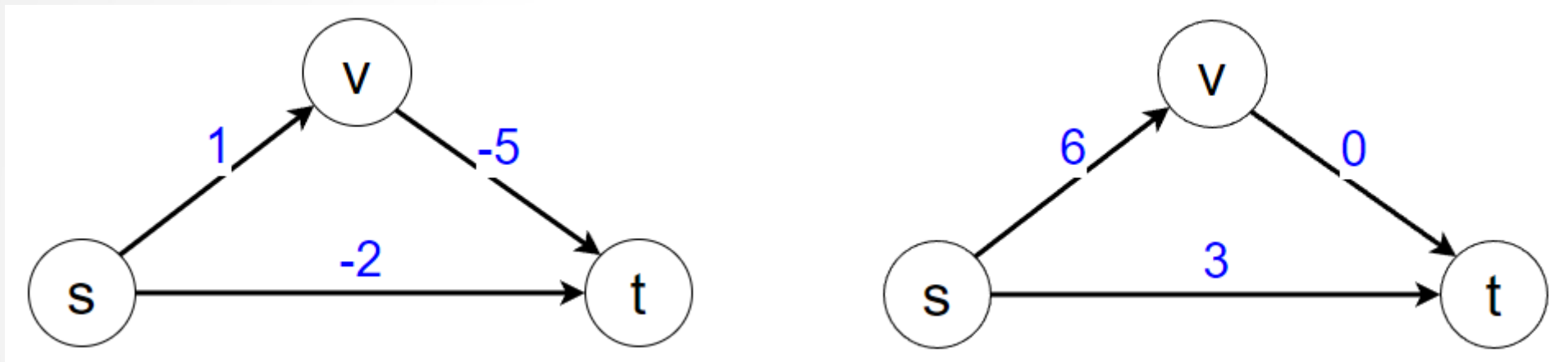
Итерация 1: $V' = \{s, t\}$, $d[t] = -2$.

Итерация 2: $V' = \{s, t, v\}$, $d[t] = -2$, $d[v] = 1$.

Правильное значение: $d[t] = -4$.

Задачи о кратчайшем пути

Можно ли заменить веса, чтобы они стали неотрицательными?



На изменённом графе оптимальный путь из s в t : $s \rightarrow t$.

На исходном графе оптимальный путь из s в t : $s \rightarrow v \rightarrow t$.

Вывод: для случая отрицательных весов нужно разрабатывать другой алгоритм ☹

Задачи о кратчайшем пути

Итак, рассмотрим общий вариант задачи: граф $G(V, E)$, $w: E \rightarrow R$.

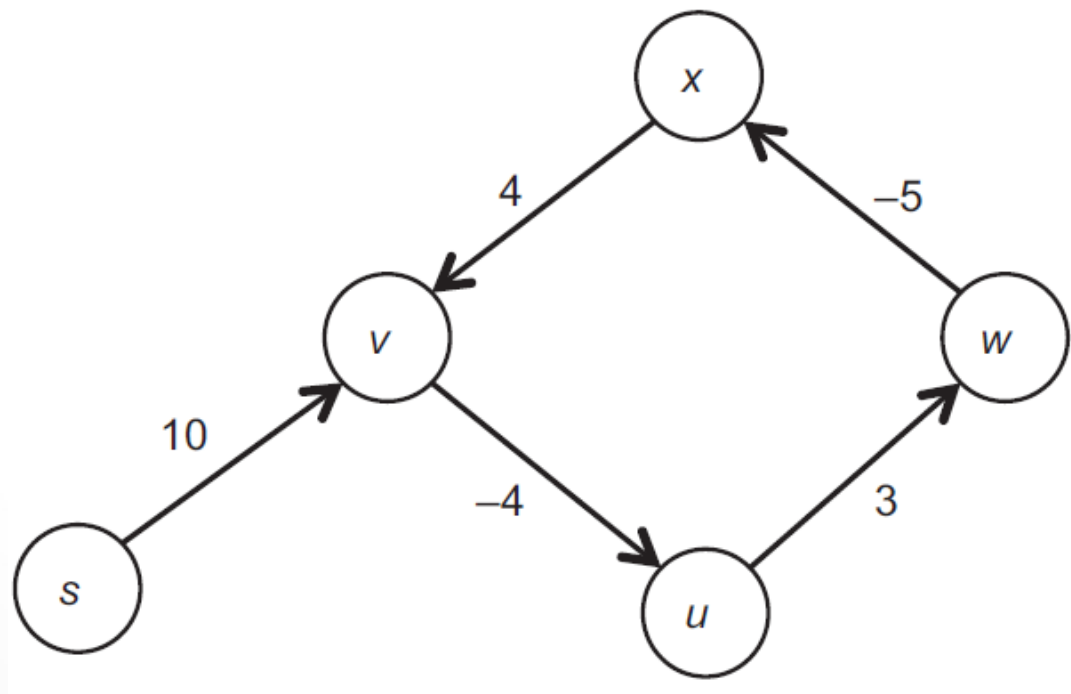
То есть, допускаются отрицательные веса дуг.

Но тогда возникает проблема с *контурами* отрицательного веса.

Каков вес кратчайшего пути

из s в v ?

- a) 10
- b) 8
- c) 6
- d) 4
- e) ?



Задачи о кратчайшем пути

Возможный вариант: искать только пути, не содержащие контуров.

К сожалению, такая постановка задачи является NP-трудной (к ней можно свести незамкнутую ориентированную задачу коммивояжера).

Поэтому поставим себе задачу в таком виде: для заданного графа $G(V, E)$, $w: E \rightarrow R$, с отрицательными весами дуг, и выбранной вершины $s \in V$ выдать один из ответов:

- a) «В графе есть контур отрицательного веса» (достижимый из s).
- b) Если контуров отрицательного веса нет, то построить кратчайшие пути из s во все вершины графа.

Эту задачу эффективно решает **алгоритм Беллмана-Форда**.