Семинар 7

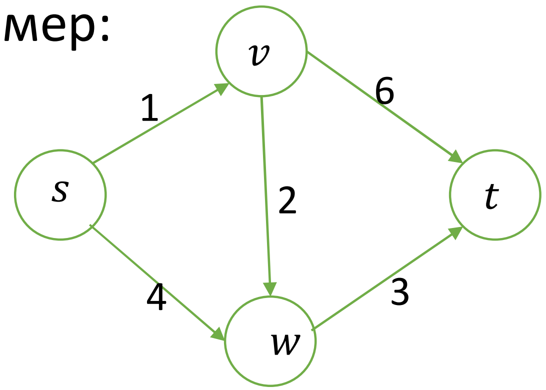
Кратчайшие пути во взвешенном графе

Взвешенный граф G(V, E)

Рассмотрим взвешенный ориентированный граф G(V, E).

Любому ребру (v, w) соответствует неотрицательное число l_{vw} (вес, длина, стоимость). Необходимо найти пути из $s \in V$ во все вершины графа так, чтобы длины этих путей были минимальны.

Пример:



Каковы кратчайшие пути до s, v, w, t ?

а) 0, 1, 2, 3

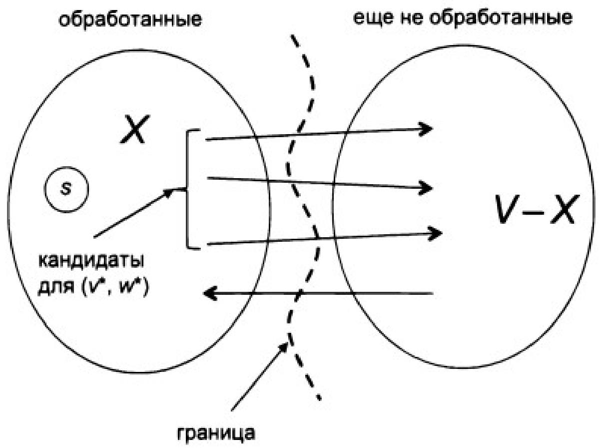
б) 0, 1, 3, 6

в) 0, 1, 4, 6

г) 0, 1, 4, 7

Почему не поиск в ширину?

Алгоритм Дейкстры



Каждая итерация алгоритма Дейкстры обрабатывает одну новую вершину, голову ребра, переходящего из X в $V-X$

DIJKSTRA

Вход: ориентированный граф $G=(V,E)$, представленный в виде списков смежности, вершина $s\in V$, длина $l_e \geq 0$ для каждого $e \in E$.

Постусловие: для каждой вершины v значение $len(v)$ равно истинному кратчайшему расстоянию $dist(s, v)$.

// инициализация

1) $X := \{s\}$
2) $len(s) := 0$, $len(v) := +\infty$ для каждого $v \neq s$

// главный цикл

3) **while** существует ребро (v, w), где $v \in X$, $w \notin X$ **do**
4) $(v^*, w^*) :=$ такое ребро, которое минимизирует $len(v) + l_{vw}$
5) $\mathrm{prev}[w^*] = v^*$
6) добавить w^* в X
7) $len(w^*) := len(v^*) + l_{v^*w^*}$.

Тестовое задание

Какое из следующих времен выполнения лучше всего описывает простую реализацию алгоритма Дейкстры для графов, представленных в виде списков смежности?

Обозначим $n = |V|, m = |E|$

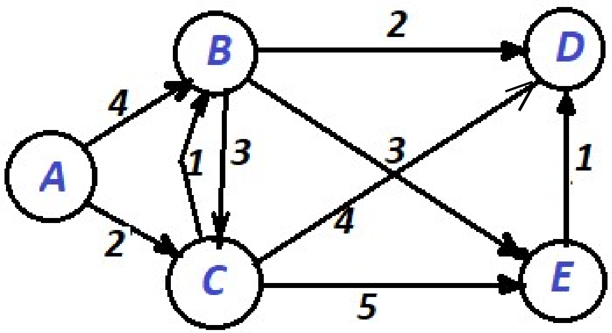
а) $O(m + n)$

б) $O(m \cdot \log n)$

в) $O(n^2)$

г) $O(m \cdot n)$

Пример.



Заполнить таблицы

X	**V-X**
A(0)	B(1), C(2), D(∞), E(∞)
…	

A	**B**	**C**	**D**	**E**	**prev**
<u>0</u>	∞	∞	∞	∞	-
<u>0</u>	4	…			

Ускорение алгоритма Дейкстры

Если на шаге 4 (4) (v^*, w^*) := такое ребро, которое минимизирует $len(v) + l_{vw}$) выбирать $\min\limits_{\substack{v \in X, \\ w \in V-X}} (len(v) + l_{vw}) = l_{v^* w^*}$

для каждой из $O(|V|)$ вершин перебирать $O(|E|)$ ребер, то время работы $O(|V| \cdot |E|)$. Можно ли быстрее? – да, если использовать двоичную кучу.

Куча поддерживает основные операции «вставить» и «извлечь минимум», которые выполняются за время $O(\log n)$.
Будем хранить в куче необработанные вершины из V-X.
При этом будем поддерживать инвариант:
Ключом вершины $w \in V - X$ является минимальная дейкстрова оценка ребра с хвостом $v \in X$ и с головой $w \in V - X$ либо $+\infty$, если такое ребро не существует:

$$key(w) = \min_{\substack{v \in X, \\ w \in V-X}} (len(v) + l_{vw})$$

Алгоритм Дейкстры на основе двоичной кучи

DIJKSTRA (НА ОСНОВЕ КУЧИ, ЧАСТЬ 1)

Вход: ориентированный граф $G = (V, E)$, представленный в виде списков смежности, вершина $s \in V$, длина $l_e \geq 0$ для каждого $e \in E$.

Постусловие: для каждой вершины v значение $len(v)$ равно истинному кратчайшему расстоянию $dist(s, v)$.

// Инициализация

1) $X :=$ пустое множество, $H :=$ пустая куча
2) $key(s) := 0$
3) **for** каждая $v \neq s$ **do**
4) $\quad key(v) := +\infty$; $prev := nil$
5) **for** каждая $v \in V$ **do**
6) $\quad$ Вставить v в H // либо использовать операцию // «Объединить в кучу»

// Главный цикл

7) **while** H является непустой **do**
8) $\quad w^* :=$ Извлечь минимум (H)
9) $\quad$ добавить w^* в X
10) $\quad len(w^*) := key(w^*)$

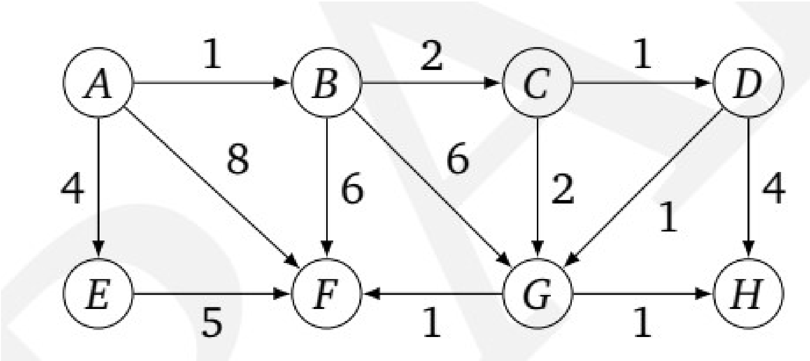
DIJKSTRA (НА ОСНОВЕ КУЧИ, ЧАСТЬ 2)

11) // обновить кучу для поддержания инварианта
12) **for** каждое ребро (w^*, y) **do**
13) $\quad$ Удалить y из H
14) $\quad key(y) := \min\{key(y), len(w^*) + l_{w^*y}\}$
15) $\quad prev := w^*$
16) $\quad$ вставить y в H

Сколько раз Dijkstra (на основе кучи) выполняет строки 13 и 16?

а) $O(m + n)$
б) $O(m)$
в) $O(n^2)$
г) $O(m \cdot n)$

Задача 1. Применить алгоритм Дейкстры к следующему графу. Привести результаты последовательных итераций в виде таблиц (см. слайд 4). Нарисовать дерево кратчайших путей



Задача 2. Докажите, что ребра $(u, prev[u])$, полученные в результате работы алгоритма Дейкстры, образуют дерево

Задача 3. Дан граф G=(V, E) с (возможно, отрицательными) весами на ребрах, а также $s \in V$ и дерево $T = (V, E'), E' \in E$. Постройте алгоритм, проверяющий за линейное время, является ли T деревом кратчайших путей.