Курс kiev-clrs – Лекция 16. Жадные алгоритмы. Графы.

Иван Веселов

2009 г.

Содержание

1 План лекции

- Представление графов
- Минимальные остовные деревья
- Оптимальная подструктура
- Жадный выбор
- Алгоритм Прима

2 Графы

Ориентированный граф (орграф) $G = (V, E)$ – это упорядоченная пара, состоящая из множества вершин V и множества рёбер $E \subseteq V \times V$

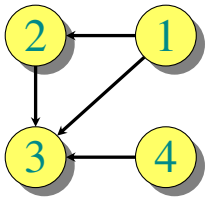
В неориентированном графе $G = (V, E)$, множество рёбер E состоит из *неупорядоченных* пар вершин.

В любом случае получаем $|E| = O(V^2)$.

Кроме того, если граф G связен, то $|E| \geqslant |V| - 1$, таким образом получаем, что $\lg |E| = \Theta(\lg V)$.

Матрица инцидентности для графа $G = (V, E)$, где $V = 1, 2, \cdots, n$:

$$A[i, j] = \begin{cases} 1 \text{ если } (i, j) \in E \\ 0 \text{ в противном случае} \end{cases}$$



A	1	2	3	4
1	0	1	1	0
2	0	0	1	0
3	0	0	0	0
4	0	0	1	0

Рис. 1: Матрица инцидентности

Это так называемое плотное заполнениие. Для разреженных матриц больше подходит списочное представление графа. Т.е. для каждой вершины v – имеем список вершин $Adj[v]$, который задаётся исходящими из вершины v рёбрами.

$$Adj[1] = \{2, 3\}$$
$$Adj[2] = \{3\}$$
$$Adj[3] = \{\}$$
$$Adj[4] = \{3\}$$

Для ориентированных графов размер этого списка называется “исходящей степенью” вершины.

Для неориентированных графов – просто “степенью”.

Лемма о рукопожатиях:

$\sum_{v \in V} degree(v) = 2|E|$ для неориентированного графа, таким образом списки смежности используют $\Theta(V + E)$ места для хранения.

3 Минимальные остовные деревья

Вход: взвешенный связный неориентированный граф $G = (V, E)$ с функцией веса $W : E \to \mathbb{R}$

Для простоты предположим, что все веса различны.

Результат: остовное дерево T, которое соединяет все вершины и обладает минимальным весом

$$w(T) = \sum_{(u,v) \in T} w(u, v)$$

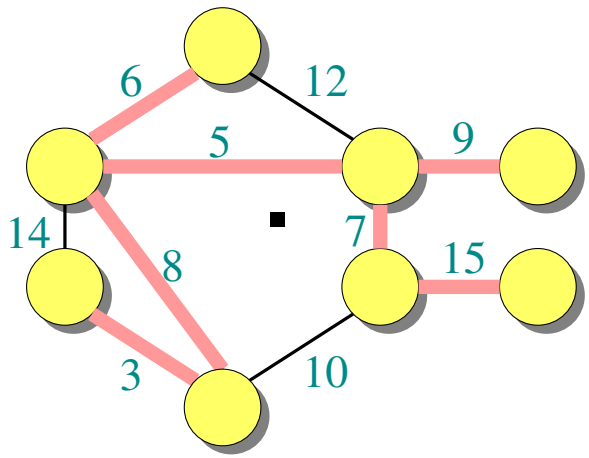


Рис. 2: Минимальное остовное дерево

Оптимальная подструктура.

Рассмотрим минимальное остовное дерево T, без указания всего остального графа G.

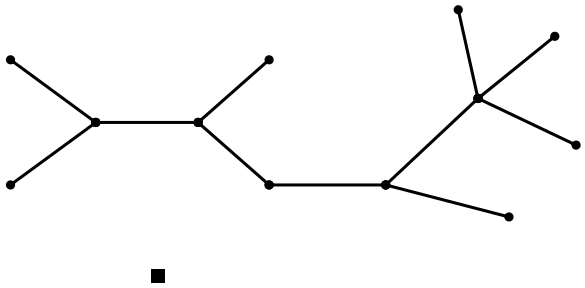


Рис. 3: Минимальное остовное дерево

Рассмотрим какое-либо ребро (u, v)

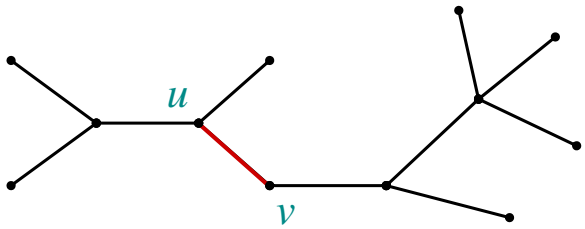


Рис. 4: Минимальное остовное дерево – с выделенным ребром (u, v)

и удалим его

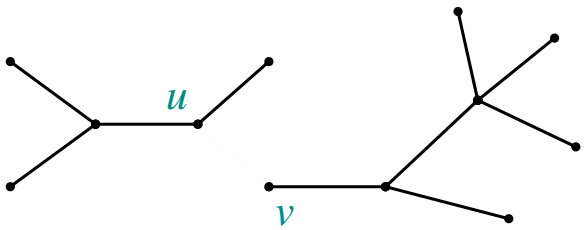


Рис. 5: Минимальное остовное дерево – после удаления ребра

После этого дерево перестанет быть связным и распадётся на два поддерева

О том, что имеет место оптимальная подструктура говорит следующая **теорема**:

Поддерево T_1, являеется МОД для графа $G_1 = (V_1, E_1)$, который является подграфом G и индуцирован по вершинам T_1, то есть включает в себя только те рёбра, что соединяют вершины V_1. Более формально:

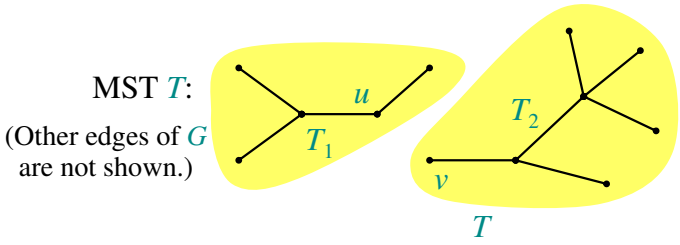


Рис. 6: Минимальное остовное дерево с двумя поддеревьями

$$V_1 = \text{вершины } T_1$$
$$E_1 = \{(x, y) \in E : x, y \in E_1\}$$

Аналогично для T_2.

Доказательство (техника "Cut and paste")

Вес МОД T:

$$w(T) = w(T_1) + w(u, v) + w(T2)$$

Пусть T_1' – является ОД для графа G_1 с меньшим весом чем T_1 Тогда мы могли бы использовать ("вставить") его в нашем оригинальном дереве T и получить вес

$$w'(T) = w(T_1') + w(u, v) + w(T2) < w(T)$$

что противоречит изначальному предположению о том, что $w(t)$ – минимален.

Получается, что у нас есть оптимальная подструктура. Есть ли пересекающиеся подзадачи? Да, т.к. нам приходится несколько раз решать одну и ту же подзадачу для какой-то части графа.

Значит мы можем использовать динамическое программирование!

Но кроме это у нашей задачи есть ещё одно важное свойство, которое позволяет решить задачу эффективнее.

Признак жадного алгоритма

Свойство жадного выбора – означает, что локально оптимальный выбор является и глобально оптимальным.

О том, что для МОД выполняется свойство жадного выбора говорит следующая **теорема**:

Пусть T – МОД для $G = (V, E)$ и пусть $A \subseteq V$. Предположим, что $(u, v) \in E$ – ребро с наименьшим весом среди рёбер соединяющих A с $V - A$. Тогда $(u, v) \in T$

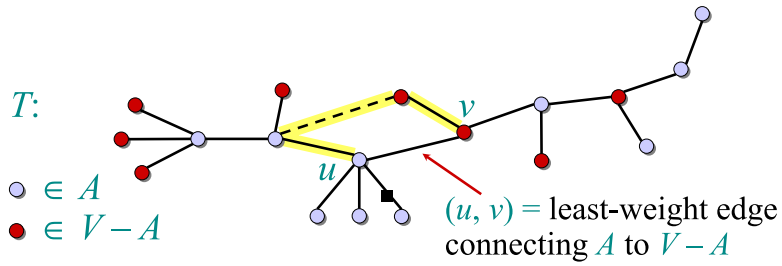


Рис. 7: Иллюстрация доказательства

Доказательство (“cut and paste”)

- пусть ребро $(u, v) \notin T$ и является ребром с наименьшим весом среди тех, что соединяют A и $V - A$.
- заметим, что должен существовать единый простой путь от u к v в нашем МОД T
- заменим (u, v) на первое ребро этого пути, которое соединяет A и $V - A$ (такое ребро всегда есть, т.к. u и v находятся в разных множествах и где-то должен быть переход).
- получим МОД с меньшим суммарным весом, что ведёт к противоречию

4 Алгоритм Прима

Идея: представить $V - A$ с помощью очереди с приоритетами Q. Каждая вершина в Q будет хранить ключ – значение наименьшего по весу ребра, соединяющего её с вершинами из A.

```
1  Q ← V
2  key[v] ← ∞ для всех v ∈ V
3  key[s] ← 0 для некоего s ∈ V
4  while Q ≠ ∅
5        do u ← Extract-Min(Q)
6           for each v ∈ Adj[u]
7                 do if v ∈ Q и w(u, v) < key(v)
8                       then key[v] ← w(u, v)
9                            π(v) ← u
```

В конце возвращаем $\{(v, \pi(v))\}$

Анализ алгоритма Прима

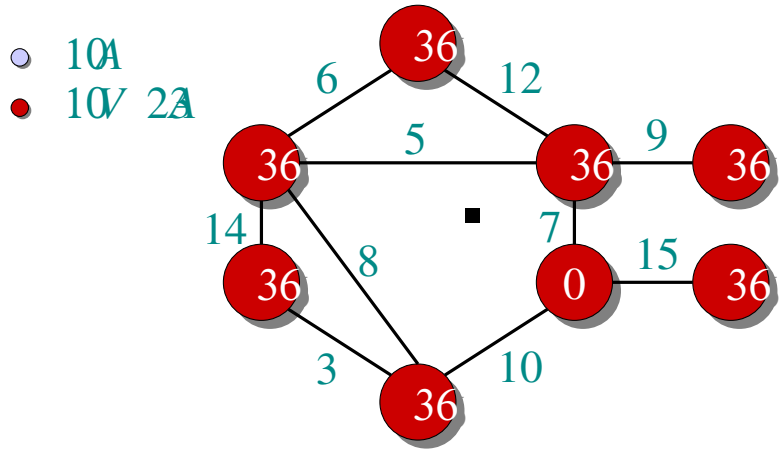


Рис. 8: Пример алгоритма Прима

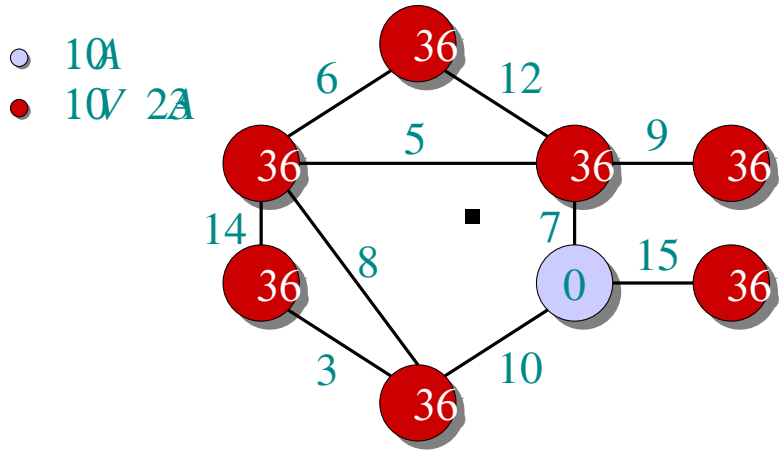


Рис. 9: Пример алгоритма Прима

Другие алгоритмы для поиска МОД

- Алгоритм Крускала (с помощью непересекающихся множеств), время выполнения $O(E \lg V)$
- Лучший известный на сегодня: рандомизированный алгоритма Каргера, Кляйна и Тарджана 1993го года, ожидаемое время выполнения $= O(V + E)$

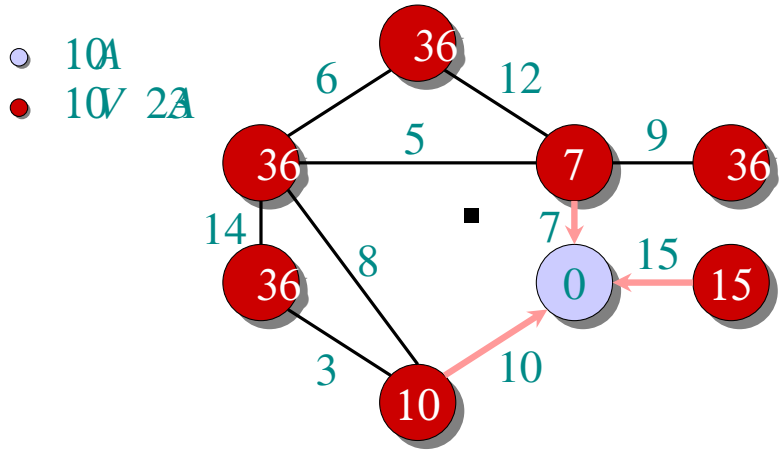


Рис. 10: Пример алгоритма Прима

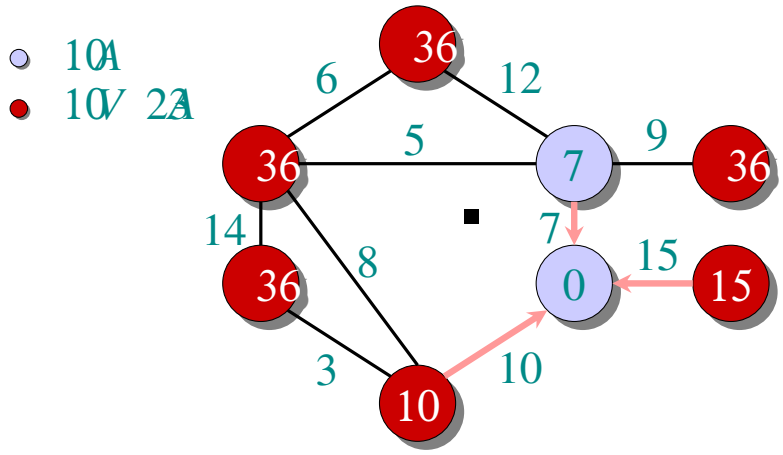


Рис. 11: Пример алгоритма Прима

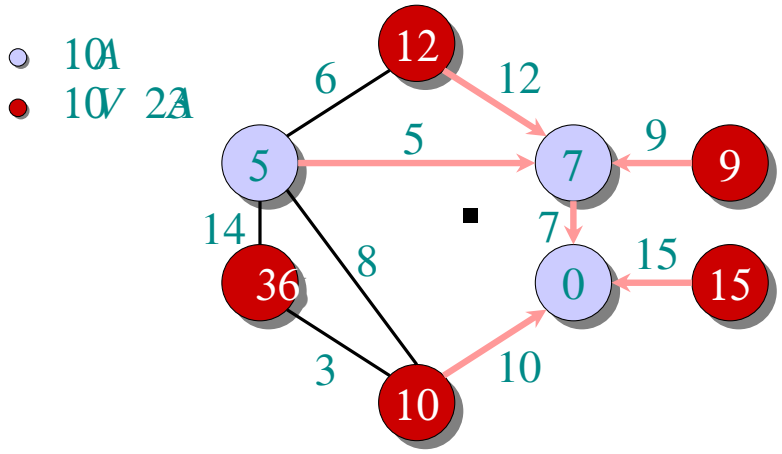


Рис. 12: Пример алгоритма Прима

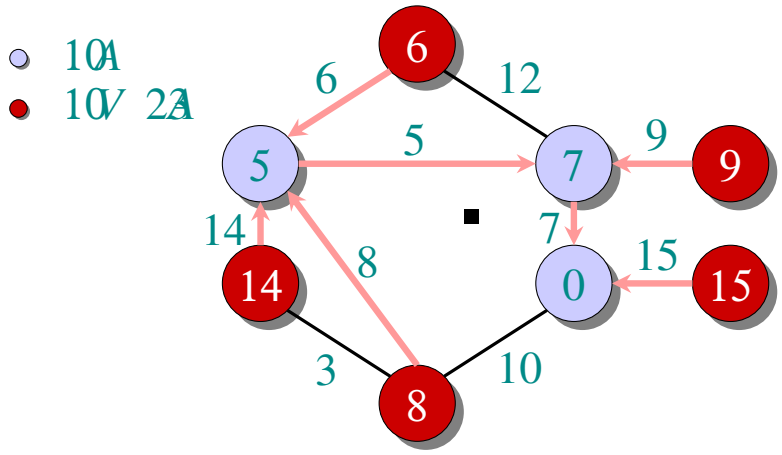


Рис. 13: Пример алгоритма Прима

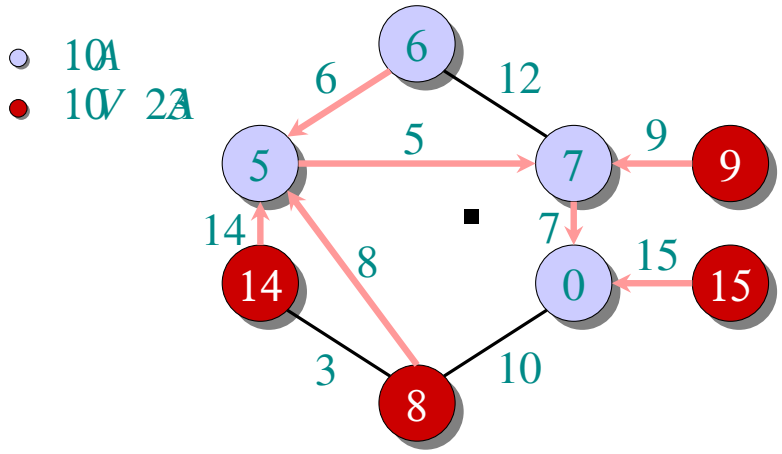


Рис. 14: Пример алгоритма Прима

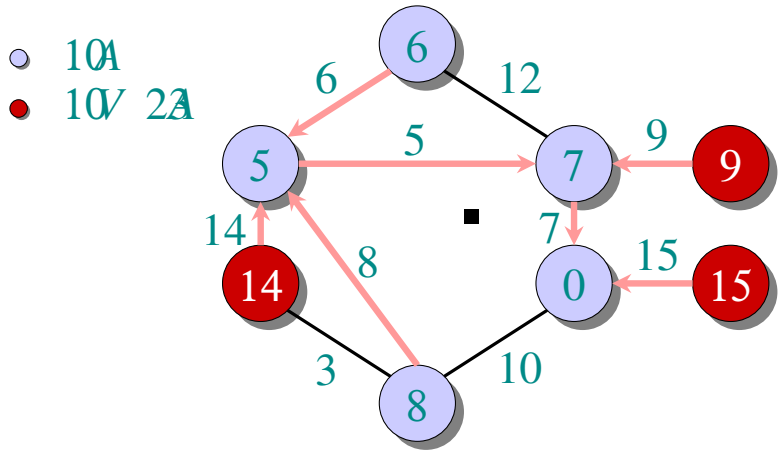


Рис. 15: Пример алгоритма Прима

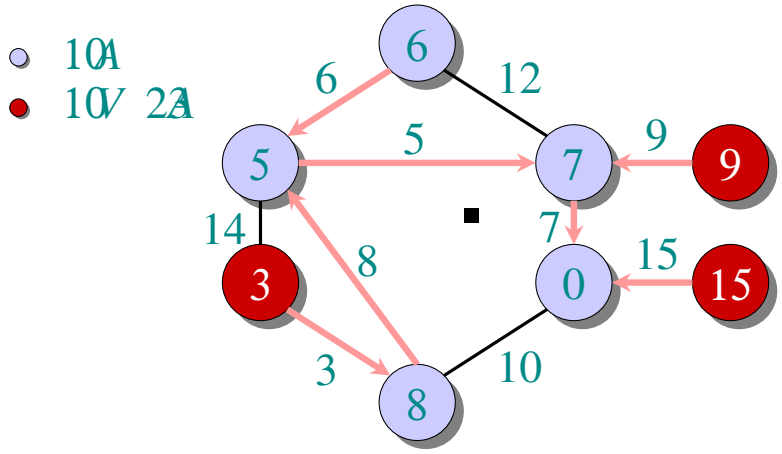


Рис. 16: Пример алгоритма Прима

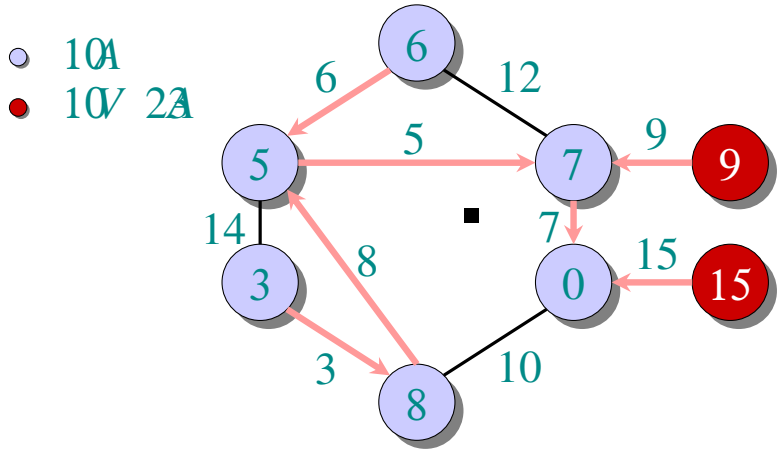


Рис. 17: Пример алгоритма Прима

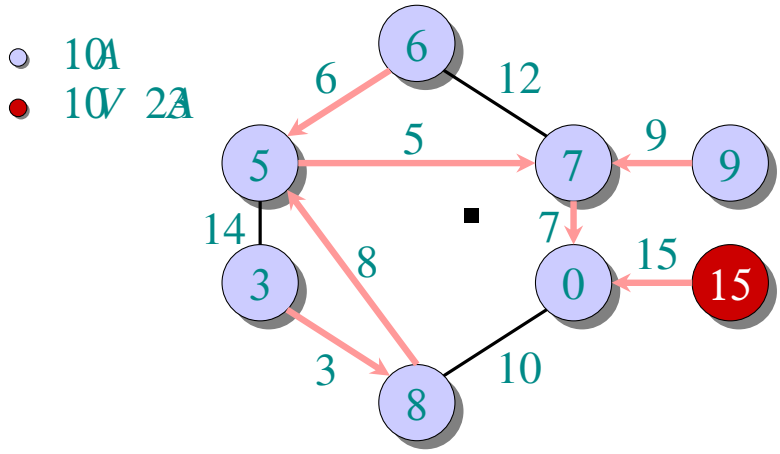


Рис. 18: Пример алгоритма Прима

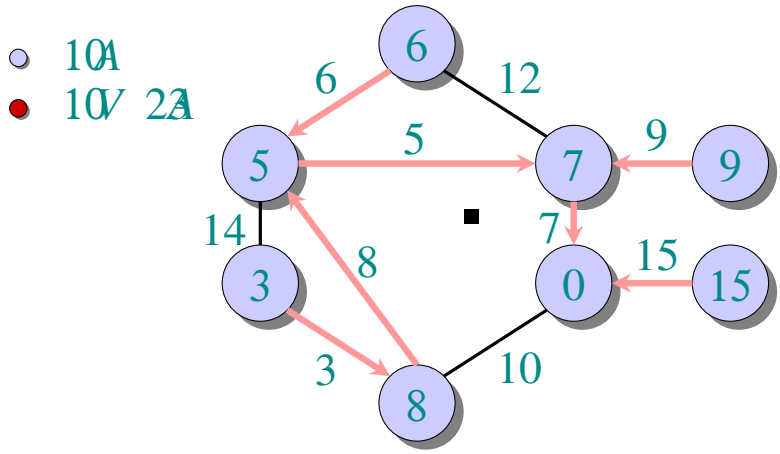


Рис. 19: Пример алгоритма Прима

$\Theta(V)$ total:
- $Q \leftarrow V$
- $key[v] \leftarrow \infty$ for all $v \in V$
- $key[s] \leftarrow 0$ for some arbitrary $s \in V$

$|V|$ times:

while $Q \neq \varnothing$

do $u \leftarrow$ EXTRACT-MIN(Q)

$degree(u)$ times:

for each $v \in Adj[u]$

do if $v \in Q$ and $w(u, v) < key[v]$

then $key[v] \leftarrow w(u, v)$

$\pi[v] \leftarrow u$

Handshaking Lemma $\Rightarrow \Theta(E)$ implicit DECREASE-KEY's.

Time $= \Theta(V) \cdot T_{\text{EXTRACT-MIN}} + \Theta(E) \cdot T_{\text{DECREASE-KEY}}$

$$\text{Time} = \Theta(V) \cdot T_{\text{EXTRACT-MIN}} + \Theta(E) \cdot T_{\text{DECREASE-KEY}}$$

Q	$T_{\text{EXTRACT-MIN}}$	$T_{\text{DECREASE-KEY}}$	Total
array	$O(V)$	$O(1)$	$O(V^2)$
binary heap	$O(\lg V)$	$O(\lg V)$	$O(E \lg V)$
Fibonacci heap	$O(\lg V)$ amortized	$O(1)$ amortized	$O(E + V \lg V)$ worst case