## *Лекція 30. Алгоритм Дейкстри*

Розглянемо наступну задачу: заданий скінченний орієнтований граф $G$, кожному ребру якого приписана його числова характеристика („довжина”). Необхідно знайти найкоротший маршрут від заданої вершини (позначимо її через $s$) до всіх решта вершин графу.

Для розв’язання цієї задачі розповсюдження набув алгоритм Дейкстри, згідно з яким всі вершини графу $G(V)$ необхідно впорядкувати за зростанням їх відстані від вершини $s$:

$$d(s, u_0) \le d(s, u_1) \le \ldots \le d(s, u_n) \ \text{ і } V = \{ u_0, u_1, \ldots, u_n\}.$$

Ця послідовність будується інтеративно. Перша вершина в ній відома:

$$u_0 = s; d(s, u_0) = 0.$$

Нехай відомі перші $(i + 1)$ вершини цієї послідовності:

$$\{u_0 = s, u_1, \ldots, u_i\},$$

які ми будемо надалі називати фіксованими. Це означає, що вершина $u_i$ ближча від $s$, ніж всі решта (тобто нефіксовані) вершини.

Для кожної нефіксованої вершини $v$ графу модифікуємо її відстань від $s$: якщо існує ребро $e(u_i, v)$ і $d(s, u_i) + d(u_i, v) \le d(s, v)$ , то $d(s, v) = d(s, u_i) + d(u_i, v)$ і передостанньою вершиною в найкоротшому (на даний час) шляху, який з’єднує $s$ з $v$, буде вершина $u_i$. Після цього серед всіх нефіксованих вершин знаходимо ту, відстань до якої від $s$ є найменшою. Ця вершина і буде наступною в послідовності (тобто $u_{i+1}$).

**Опис алгоритму.**

Масиви, що використовуються:

VID: VID($i$) – найкоротша на даний момент відстань від $s$ до $i$ -ї вершини графу;

FIX: FIX($i$) = 1, якщо $i$-та вершина графу є фіксованою;

PERED: PERED($i$) містить передостанню вершину в найкоротшому з усіх відомих на даний момент маршрутів від $s$ до $i$-ї вершини графу.

1) Початкові встановлення.

Для початкової вершини: VID($s$) = 0, FIX($s$) = 1, PERED($s$) = $s$.

Для всіх інших вершин графу: VID($v$) = ∞, FIX($v$) = 0, PERED($v$) = $v$.

Поточна вершина: $u = s$.

Номер етапу: $j = 1$.

2) ЦИКЛ по тих вершинах графу $G$, для яких $\mathrm{FIX}(v) = 0$

ЯКЩО існує $e = (u, v)$ і $\mathrm{VID}(u) + d(u, v) \le \mathrm{VID}(v)$

ТО $\mathrm{VID}(v) = \mathrm{VID}(u) + d(u, v)$, $\mathrm{PERED}(v) = u$.

3) Серед вершин графу $G$, для яких $\mathrm{FIX}(v) = 0$, знаходимо ту вершину $v_0$, для якої

$$\mathrm{VID}(v_0) = \min_{v} \mathrm{VID}(v) .$$

4) $\mathrm{FIX}(v_0) = 1$, $u = v_0$.

5) $j = j + 1$.

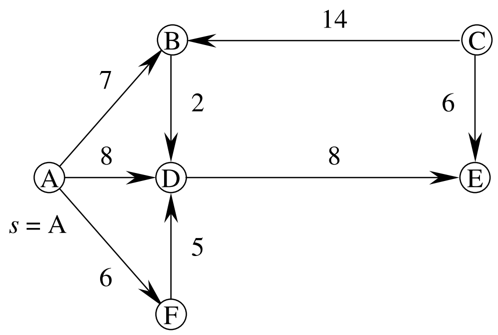
ЯКЩО $j \le n$,

ТО йти на 2).

6) Кінець.

В результаті масив VID містить найкоротші відстані від $s$ до всіх вершин графу; по масиву PERED можна отримати найкоротший маршрут від $s$ до довільної вершини графу.

Приклад.




Робота алгоритму проілюстрована в таблиці, в якій кожний рядок відповідає одному етапу роботи алгоритму. Фіксовані вершини підкреслені, а біля вершини „*u*" на кожному кроці стоїть зірочка.

| $j$ | VID | | | | | | PERED | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | A | B | C | D | E | F | A | B | C | D | E | F |
| 1 | 0* | ∞ | ∞ | ∞ | ∞ | ∞ | A | B | C | D | E | F |
| 2 | 0 | 7 | ∞ | 8 | ∞ | 6* | A | A | C | A | E | A |
| 3 | 0 | 7* | ∞ | 8 | ∞ | 6 | A | A | C | A | E | A |
| 4 | 0 | 7 | ∞ | 8* | ∞ | 6 | A | A | C | A | E | A |
| 5 | 0 | 7 | ∞ | 8 | 16* | 6 | A | A | C | A | D | A |
| 6 | 0 | 7 | ∞ | 8 | 16 | 6 | A | A | C | A | D | A |

Найкоротший маршрут від *A*, наприклад до *E*, будуємо, використовуючи масив PERED, таким чином: $E \leftarrow D \leftarrow A$.