## §3. Сети. Алгоритм Дейкстры поиска расстояний от вершины до всех остальных.

Опр. Сетью называется орграф G, с весовой функцией $\mu$ на дугах ($\mu(e)$ – неотрицательное число).

Синонимы: вес дуги, длина дуги, пропускная способность дуги.

В сетях используют другое определение полустепеней исхода и захода.

Опр. Полустепень исхода вершины $u$ – число $\rho_{out}(u)$ равное сумме весов дуг, выходящих из $u$.

Полустепень захода вершины $u$ – число $\rho_{in}(u)$ равное сумме весов дуг, входящих в $u$.

Утверждение. Сумма полустепеней исхода всех вершина графа равна сумме полустепеней захода всех вершин.

Опр. Длиной пути называется сумма весов дуг в пути.
Расстоянием от вершины $u$ до вершины $v$ называется длина кратчайшего $(u - v)$-пути.

Обозначение: $\mathrm{d}(u, v)$.
Если $(u - v)$-пути не существует, то $\mathrm{d}(u, v) = \infty$.

Задача поиска расстояний от фиксированой вершины до всех остальных.

Вспомогательные обозначения для описания алгоритма Дейкстры.

$v_0$ – фиксированная вершина.

$v_1, v_2, \ldots, v_{n-1}$ – остальные вершины.

$d_i = d(v_0, v_i)$, для всех $i = 1, \ldots, n-1$.

Определим двухместную функцию для пар вершин:

$$a(v, v') = \begin{cases} \mu(v, v'), & \text{если } (v, v') \in E \\ \infty, & \text{если } (v, v') \notin E \end{cases}.$$

Алгоритм Дейкстры.

1) Собрать множество $S = V \setminus \{v_0\}$.

Присвоить $d_i = a(v_0, v_i)$ для всех $i = 1, \ldots, n-1$.

2) Если $|S| > 1$, то выбрать в $S$ вершину $v_k$, для которой $d_k$ наименьшая (среди всех вершин в $S$).

3) Собрать множество $S = S \setminus \{v_k\}$.

Присвоить $d_i = \min(d_i, d_k + a(v_k, v))$ для всех $v \in S$. Перейти к 2).

4) Если $|S| = 1$ – остановить алгоритм.

Выход – $d_1, d_2, \ldots, d_{n-1}$.

Пример.

Теорема (без док-ва).
Массив чисел $d_1, d_2, ..., d_{n-1}$ на выходе алгоритма – набор кратчайших расстояний от $v_0$ до $v_1, v_2, ..., v_{n-1}$.

Сложность по времени алгоритма Дейкстры имеет порядок $O(n^2 + m)$ $= O(n^2)$
Сложность по памяти – ?.

§4. Потоки в сетях. Алгоритм поиска максимального потока в сети.

Опр. Источником в сети (G, μ) называется вершина $u$, для которой $\rho_{in}(u) = 0$, $\rho_{out}(u) > 0$.

Стоком в сети (G, μ) называется вершина $u$, для которой $\rho_{in}(u) > 0$, $\rho_{out}(u) = 0$.

В данном параграфе используются сети с единственным источником и единственным стоком.

Пусть сеть (G, μ) имеет единственный источник *a* и единственный сток *b*.

Опр. Потоком в сети называется функция $\varphi$ определенная на дугах графа и принимающая неотрицательные действительные значения, такая, что выполнены условия:

1) $\varphi$ (e) ≤ μ(e) для каждой дуги $e \in E$;

2) $\sum_{(u,v)\in E} \varphi(u,v) = \sum_{(w,u)\in E} \varphi(w,u)$ для каждой вершины *u*, кроме источника и стока.

Потоком дуги е называется $\varphi$ (е).

Величиной потока называется $\sum_{(a,v)\in E} \varphi(a,v) = \sum_{(u,b)\in E} \varphi(u,b)$.

Задача: для заданной сети найти максимальный поток, т.е. поток с максимальной величиной.

Пример.
Компьютерная сеть.

Опр. Пусть сеть (G, μ) имеет единственный источник $a$ и единственный сток $b$.
Разрезом называется множество дуг E′, такое, что любой путь из $a$ в $b$ содержит хотя бы одну дугу из E′.
Пропускной способностью разреза E′ называется $\sum_{e \in E'} \mu(e)$.
Разрез называется минимальным, если его пропускная способность минимальна.

Теорема (Форда-Фалкерсона, без док-ва).
В любой сети величина максимального потока равна пропускной способности минимального разреза.

Пример применения теоремы Форда- Фалкерсона.

Замечание: если в сети есть дуги, вес которых – рациональное нецелое число, то задачу поиска максимального потока можно переформулировать для сети с целочисленными весами.

Пример.

Алгоритм поиска максимального потока в целочисленной сети (G, $\mu$) с источником $a$ и стоком $b$.

1) Присвоить $\varphi(e) = 0$ для каждой дуги $e$.
Собрать T = {все пути из $a$ в $b$, не имеющие общих дуг}.
Если T = $\varnothing$, то закончить алгоритм.
Если T $\neq \varnothing$, то для каждого пути $e_1, e_2, \ldots, e_k$ присвоить
$\varphi(e_1) = \varphi(e_2) = \ldots = \varphi(e_k) = \min(\mu(e_1), \mu(e_2), \ldots, \mu(e_k))$.

2) Построить сеть (G′, μ′), с множеством вершин V и множеством дуг E′: если $e \in E$ и (μ(e) – $\varphi(e)$) > 0, то $e \in E'$, μ′(e) = (μ(e) – $\varphi(e)$).
Для каждой дуги $e \in E$, у которой $\varphi(e)$ > 0, добавляем в E′ «обратную к *e*» дугу $e_{re}$, $\mu'(e_{re}) = \varphi(e)$.
При появлении кратных дуг, заменяем на одну дугу, вес которой равен сумме весов кратных дуг.
Для сети (G′, μ′) построить поток $\varphi'$ как в шаге 1).
3) Присвоить $\varphi = \varphi + \varphi'$. Вернуться к шагу 2).

Пример.