# Пример поиска в ширину

*Search1\_B;*

/\* Модификация алгоритма волнового фронта для произвольных неотрицательных числовых меток дуг. IN – признак присутствия вершины в очереди. В отличие от классического алгоритма волнового фронта вершина может многократно попадать в очередь. \*/

**for all** v ∈ V **do** M[v]:= ∞; IN[v]:= 0; **end**

Q:= ∅;

M[v0]:= 0;

v0 🡪 Q; IN[v0]:=1;

**for all** v ∈ Q **while** (Q ≠ ∅) **do**

**for all** w ∈ L[v] **do**

Δ:= min(M[w], M[v] + ϕ(<v, w>));

**if** ( Δ < M[w]) **then begin**

M[w]:= Δ;

**if** IN[w]=0 **then begin**

w 🡪 Q;

IN[w]:= 1;

**end**

**end**

**end**

Q 🡪 v; IN[v]:= 0;

**end**

**end** *Proc Search1\_B*

Рассмотрим поиск в ширину в графе, размеченном над полукольцом (рис. 1).

Матрица: 

1

1

1

3

5

2

1🡪2🡪3🡪4

5🡪6

В данном случае формулы перевычисления меток имеют вид:



Процесс работы алгоритма сведем в таблицу 1.

В этой таблице в колонке, описывающей изменения содержимого очереди, точкой с запятой мы отделяем текущую вершину поиска, пока еще находящуюся в очереди.

**Таблица 1**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Очередь Q | M[1] | M[2] | M[3] | M[4] | M[5] | M[6] |
| 2 | 1 | 0 | ∞ | ∞ | ∞ | ∞ | ∞ |
| 3 | 1; 2, 5 | 0 | 2 | ∞ | ∞ | 1 | ∞ |
| 4 | 2; 5, 3 | 0 | 2 | 7 | ∞ | 1 | ∞ |
| 5 | 5;3, 6 | 0 | 2 | 7 | ∞ | 1 | 2 |
| 6 | 3; 6, 4 | 0 | 2 | 7 | 10 | 1 | 2 |
| 7 | 6; 4, 3 | 0 | 2 | **3** | 10 | 1 | 2 |
| 8 | 4;3 | 0 | 2 | 3 | 10 | 1 | 2 |
| 9 | 3 | 0 | 2 | 3 | **6** | 1 | 2 |
| 10 | 4 | 0 | 2 | 3 | 6 | 1 | 2 |
| 11 | Ø | 0 | 2 | 3 | 6 | 1 | 2 |

В строке 7 показано уменьшение метки вершины 3, и, поскольку она уже покинула очередь, ее необходимо туда вернуть. В строке 9 показано уменьшение метки вершины 4, и она тоже второй раз становится в очередь, но тут же покидает ее, так как множество ее преемников пусто.

Строка 1 показывает состояние до выполнения основного цикла.