10.4. Dijkstra's Algorithm

By: Snezhana Gocheva-Ilieva, snow@uni-plovdiv.bg

Basic problem 3

An oriented graph is given, the lengths of the arcs of which are all nonnegative, i.e. $R_{ij} \geq 0$. Find the minimal paths from a given node to all other nodes of the graph.

This problem is a special case of problem 2, which is solved in the previous paragraph and can be solved using Bellman-Ford's algorithm. In this case we offer a more efficient algorithm, requiring only $O(n^2)$ arithmetical operations, also known as Dijkstra's algorithm.

Formulas for Dijkstra's algorithm for a graph with *n* nodes

Let's consider for convenience the starting node as number 1. According to standards we V stands for the totality of nodes of the graph. We introduce another set T which at the beginning is equal to the set of nodes {2, 3, … , n}, i.e. $T=V \setminus \{1\}$. We calculate in stages and write down the results in a table just like with Bellman-Ford's method. At every stage (row of the table) a main element is chosen from the remaining nodes T, equal to a min of D_i, with u standing for the number of the node for which a minimum is reached. At the next stage this knot is excluded from T because the distance for this point has reached its possible minimum. We continue this procedure until all nodes are excluded, i.e. until $T = \varnothing$.

k =1. $D_i = R_{1i}$, i= 1, 2, … , n.

k =2. A main element is chosen $D_u = \min \{D_i, i \in T\}$, $T = T \setminus \{u\}$. For all points from T we calculate

$$D_2 = \min \{D_2, D_u + R_{u,2}\},$$

(2) $$D_3 = \min \{D_3, D_u + R_{u,3}\},$$

$$\ldots$$

$$D_n = \min \{D_n, D_u + R_{u,n}\}.$$

k =3, 4, ... , n-1.

<u>Example 5.</u> Let's solve the problem from example 4 using Dijkstra's algorithm.

Solution. The distances matrix is

$$R = \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} \begin{bmatrix} 0 & 1 & \infty & \infty & \infty & \infty \\ \infty & 0 & 5 & 2 & \infty & 7 \\ \infty & \infty & 0 & \infty & \infty & 1 \\ 2 & \infty & 1 & 0 & 4 & \infty \\ \infty & \infty & \infty & 3 & 0 & \infty \\ \infty & \infty & \infty & \infty & 1 & 0 \end{bmatrix},$$

k =1, D_1= 0, $D_2 = R_{12}$ = 1, $D_3 = D_4 = D_5 = D_6 = \infty$. T = {2, 3, 4, 5, 6}.

k =2. A main element is chosen D_u = min $\{D_i, i \in T\}$ = 1, i.e. u =2, D_u = 1; T = {3, 4, 5, 6}. In the table the main element has been circled.

D_3 = min $\{D_3, D_2 + R_{2,3}\}$ = min $\{\infty, \underline{1+5}\}$ = 6,
D_4 = min $\{D_4, D_2 + R_{2,4}\}$ = min $\{\infty, \underline{1+2}\}$ = 3,
D_5 = min $\{D_5, D_2 + R_{2,5}\}$ = min $\{\infty, \underline{1+\infty}\} = \infty$,
D_6 = min $\{D_6, D_2 + R_{2,6}\}$ = min $\{\infty, \underline{1+7}\}$ = 8.

k =3. A main element is chosen D_u = min $\{D_i, i \in T\}$ = 3, i.e. u =4, D_u = 3; T = {3, 5, 6}.
D_3 = min $\{D_3, D_4 + R_{4,3}\}$ = min $\{6, \underline{3+1}\}$ = 4,
D_5 = min $\{D_5, D_4 + R_{4,5}\}$ = min $\{\infty, \underline{3+4}\}$ = 7,
D_6 = min $\{D_6, D_4 + R_{4,6}\}$ = min $\{\underline{8}, 3+\infty\}$ = 8.

k =4. A main element is chosen D_u = min $\{D_i, i \in T\}$ = 4, i.e. u =3, D_u = 4; T = {5, 6}.
D_5 = min $\{D_5, D_3 + R_{3,5}\}$ = min $\{\underline{7}, 4+\infty\}$ = 7,
D_6 = min $\{D_6, D_3 + R_{3,6}\}$ = min $\{8, \underline{4+1}\}$ = 5.

k =5. A main element is chosen D_u = min $\{D_i, i \in T\}$ = 5, i.e. u =6, D_u = 5; T = {6}.
D_5 = min $\{D_5, D_6 + R_{6,5}\}$ = min $\{7, \underline{5+1}\}$ = 6.

The values circled are the sought minimal distances from node 1 to other nodes.

k	D_1	D_2	D_3	D_4	D_5	D_6	T
1	0	(1)	∞	∞	∞	∞	{2,3,4,5,6}
2		1	6	(3)	∞	8	{3,4,5,6}
3			(4)		7	8	{3,5,6}
4					7	(5)	{5,6}
5					(6)		{5}

Description of Dijkstra’s algorithm using metalanguage

An oriented weighted graph is given (V, E) with n nodes V and arcs E, which does not have negative weights. Find the minimal distances from node numbered p to all the other nodes of the graph.

The distances matrix R is given and ongoing minimal distances are recorded in an array D. $D[u]$ marks the main element, u is the number of its corresponding node. The set of unoptimized nodes up to a given stage is T. Incoming and outgoing operations have been skipped.

```
begin
for v∈V do D[v]:=R[p,v]; D[p]:=0;
T =V \{p};
While T ≠ ∅ do
   begin u:= random node w∈T, for which D[w] = min{D[q], q∈T };
          T: =T \{u};
          for v∈T do D[v]:= min ( D[v], D[u] + R[u,v] )
   end
end
```