10.5.2 Dijkstra's Algorithm

The *Dijkstra's algorithm* is an algorithm for finding the shortest paths between nodes in a graph, which may represent (for example, road networks). It was conceived by computer scientist *Edsger W. Dijkstra* in 1956. Dijkstra's original variant found the shortest path between two nodes, but a more common variant fixes a single node as the "source" vertex and finds shortest paths from the source to all other vertices in the graph, producing a shortest-path tree. Given a path P from vertex s to vertex t in a weighted graph G, we define the length of P to be the sum of the weights of its edges.

The steps involved in Dijkstra's Algorithm are as follows:

1. Set $\lambda(s) = 0$ and for all vertices $v \neq s$, $\lambda(v) = \infty$. Set $T = V$, the vertex set of G. (We will think of T as the set of vertices *uncoloured*.)

2. Let u be a vertex in T for which $\lambda(u)$ is the minimum.
3. If $u = t$, stop.
4. For every edge $e = uv$ incident with u, if $v \in T$ and $\lambda(v) > \lambda(u) + w(uv)$, change the value of $\lambda(v)$ to $\lambda(u) + w(uv)$ (That is, given an edge $e = uv$ from an *uncoloured* vertex v to u, change $\lambda(v)$ to $\min\{\lambda(v), \lambda(u) + w(uv)\}$.
5. Change T to $T - \{u\}$ and go to Step-2, (That is, *colour* u and then go back to Step-2 to find an *uncoloured* vertex with the minimum label).

Consider the weighted graph in Figure 10.17.

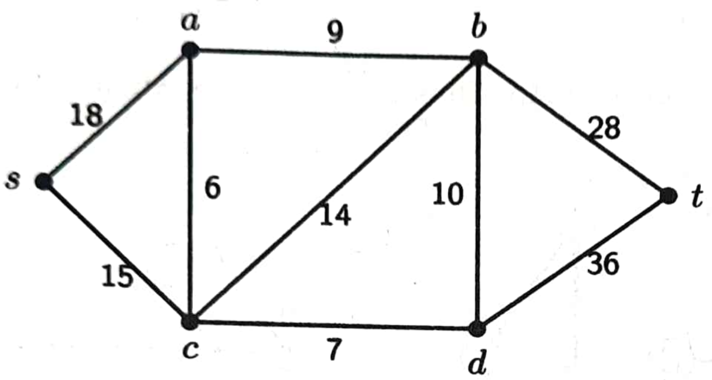

Figure 10.17: A weighted graph G

A step by step illustration of Dijkstra's Algorithm for the graph given in Figure ?? is explained as follows:

Step 1. The initial labeling is given by:

vertex v	s	a	b	c	d	t
$\lambda(v)$	0	∞	∞	∞	∞	∞
T	{s,	a,	b,	c,	d,	t}

Step 2. $u = s$ has $\lambda(u)$ a minimum (with value 0).

Step 4. There are two edges incident with u, namely sa and sc. Both a and c are in T, i.e., they are not yet coloured. $\lambda(a) = \infty > 18 = 0 + 18 = \lambda(s) + w(sa)$. Hence, $\lambda(a)$ becomes 18. Similarly, $\lambda(c)$ becomes 15.

Step 5. T becomes $T - \{s\}$, i.e., we colour s. Thus we have

vertex v	s	a	b	c	d	t
$\lambda(v)$	0	18	∞	15	∞	∞
T	{	a,	b,	c,	d,	t}

Step 2. $u = c$ has $\lambda(u)$ a minimum and u in T (with $\lambda(u) = 15$).

Step 4. There are 3 edges cv incident with $u = c$ having v in T, namely ca, cb, cd. Considering the edge ca, we have $\lambda(a) = 18 < 21 = 15 + 6 = \lambda(c) + w(ca)$ and hence $\lambda(a)$ remains as 18. Considering the edge cb, we have $\lambda(b) = \infty > 29 = 15 + 14 = \lambda(c) + w(cb)$. So, $\lambda(b)$ becomes 29. Similarly, considering the edge cd, we have $\lambda(d)$ becomes $15 + 7 = 22$.

Step 5. T becomes $T - \{c\}$, i.e., we colour c. Thus, we have

vertex v	s	a	b	c	d	t
$\lambda(v)$	0	18	29	15	22	∞
T	$\{$	$a,$	$b,$		$d,$	$t\}$

Step 2. $u = a$ has $\lambda(u)$ a minimum and for u in T (with $\lambda(u) = 18$).

Step 4. There is only one edge av incident with $u = a$ having v in T, namely ab. Then, $\lambda(b) = 29 > 27 = 18 + 9 = \lambda(a) + w(ab)$. So, $\lambda(b)$ becomes 27.

Step 5. T becomes $T - \{a\}$, i.e., we colour a. Thus, we have

vertex v	s	a	b	c	d	t
$\lambda(v)$	0	18	27	15	22	∞
T	$\{$		$b,$		$d,$	$t\}$

Step 2. $u = d$ has $\lambda(u)$ minimum for u in T (with $\lambda(u) = 22$).

Step 4. There are two edges dv incident with $u = d$ having v in T, namely db and dt. Considering the edge db, we have $\lambda(b) = 27 < 32 = 22 + 10 = \lambda(d) + w(db)$ and hence $\lambda(b)$ remains as 27. For the edge dt, we have $\lambda(t) = \infty > 58 = 22 + 36 = \lambda(d) + w(dt)$. Hence, $\lambda(t)$ becomes 58.

Step 5. T becomes $T - \{d\}$, i.e., we colour d. Thus, we have

vertex v	s	a	b	c	d	t
$\lambda(v)$	0	18	27	15	22	58
T	$\{$		$b,$			$t\}$

Step 2. $u = b$ has $\lambda(u)$ minimum for u in T (with $\lambda(u) = 27$).

Step 4. There is only one edge bv for v in T, namely bt. $\lambda(t) = 58 > 55 = 27 + 28 = \lambda(b) + w(bt)$ so $\lambda(t)$ becomes 55.

Step 5. T becomes $T - ab$, i.e., we colour b. Thus, we have

vertex v	s	a	b	c	d	t
$\lambda(v)$	0	18	27	15	22	55
T	$\{$					$t\}$

Step 2. $u = t$, the only choice. Step 3. Stop.

All steps, we did above can be written in a single table as follows:

Steps	s	a	b	c	d	t	T					
							s	a	b	c	d	t
1	0	∞	∞	∞	∞	∞	s	a	b	c	d	t
2	0	18	∞	15	∞	∞		a	b	c	d	t
3	0	18	29	15	22	∞		a	b		d	t
4	0	18	27	15	22	∞			b		d	t
5	0	18	27	15	22	58			b			t
6	0	18	27	15	22	55						t

Table 10.3: Combined table of Dijkstra's Procedure.