Algorithm Dijkstra (G, s).

// input: A weighted graph $G = (V, E)$, vertex s.

// output: The length d_v of the shortest Path from s to v and its Penultimate vertex P_v for every vertex v in V

Initialize (Q)

for every vertex v in V

$d_v \leftarrow \infty$, $P_v \leftarrow$ null

insert (Q, v, d_v)

$d_s \leftarrow 0$; Decrease (Q, s, d_s) // update priority of s with d_s

$V_T \leftarrow \phi$

for $i \leftarrow 0$ to $|V| - 1$ do

$u^* \leftarrow$ DeleteMin (Q) // delete the minimum Priority element.

$V_T \leftarrow V_T \cup \{u^*\}$

for every vertex u in $V - V_T$ that is adjacent to u^* do

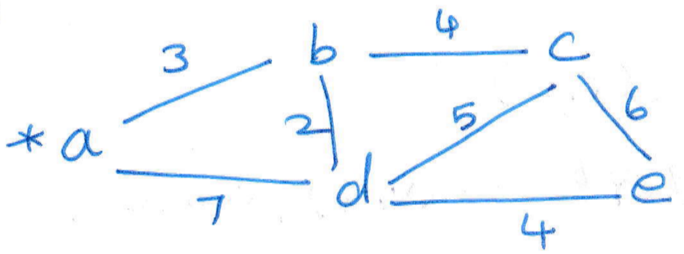
if $d_{u^*} + w(u^*, u) < d_u$

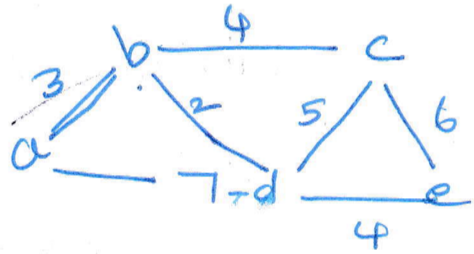
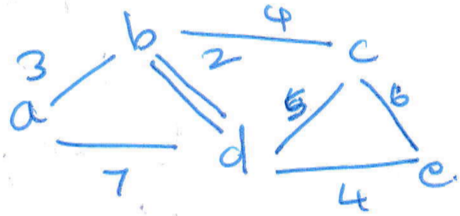
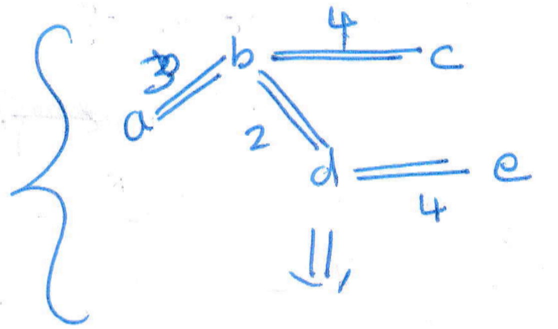
$d_u \leftarrow d_{u^*} + w(u^*, u)$

$P_u \leftarrow u^*$

Decrease (Q, u, d_u)

19



Tree vertices	Remaining vertices	illustration
a(–, 0)	b(a, 3) c(–, ∞) d(a, 7) e(–, ∞)	
b(a, 3)	c(b, 3+4) d(b (3+2), e(–∞)	
d(b, 5)	c(b, 4) e(d, 5+4)	
c(b, 4)	e(d, 9)	

		length.
from a — b		3
from a to d :	a – b – d	5
from a to c :	a – b – c	7
from a to e :	a – b – d – e	9

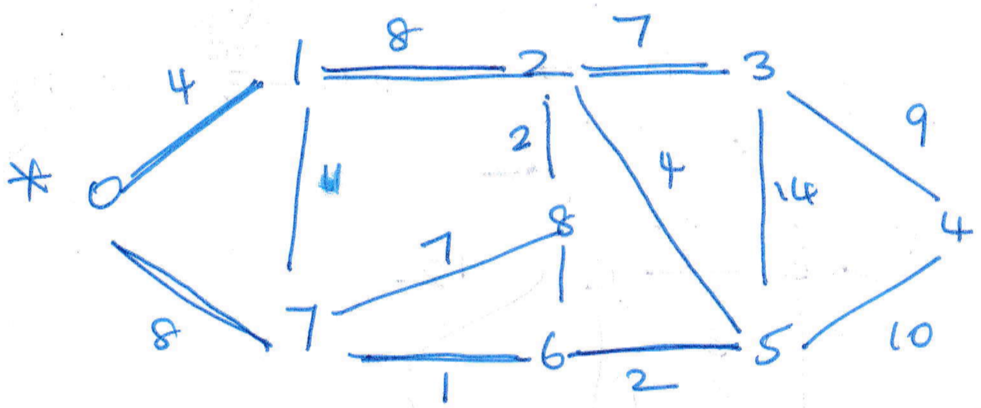
$T(n) = \Theta(v^3)$

Two operation.

* move u^* from the fringe to the set of tree vertices.

* for each remaining fringe vertex u that is connected to u^* by an edge of weight $w(u^*, u)$ such that $d_{u^*} + w(u^*, u) < d$
update the labels of u by u^* and
$d_{u^*} + w(u^*, u)$

Example Problems :.



distance from 0 to 1 is 4
0 to 2 is 0 — 1 — 2 = 12
0 to 3 is 0 — 1 — 2 — 3 = 19
0 to 4 is 0 — 7 — 6 — 5 — 4 = 2
0 to 5 is 0 — 7 — 6 — 5 = 1
6 is 0 — 7 — 6 = 9
7 is 0 ⟶ 7 = 8
8 is 0 — 1 — 2 — 8 14