Note Title

Dijkstra's Algorithm.

D[v]

p[v]

$O(n^2+m)$

Find Min $\{ D(v) \mid v \in V-S \}$

Delete Min.

$\{ D[v] \mid v \in V-S \}$.

$D[u] = D[v] + w(v,u)$

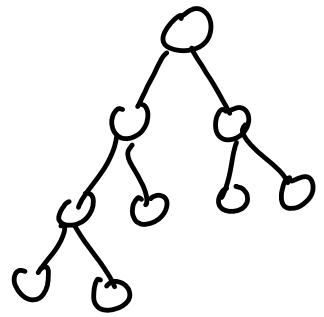
if $D[u] > D[v] + w(v,u)$

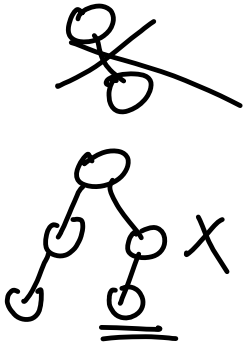
$\Rightarrow$ Every update of a $D[u]$ value involves in decreasing the value.

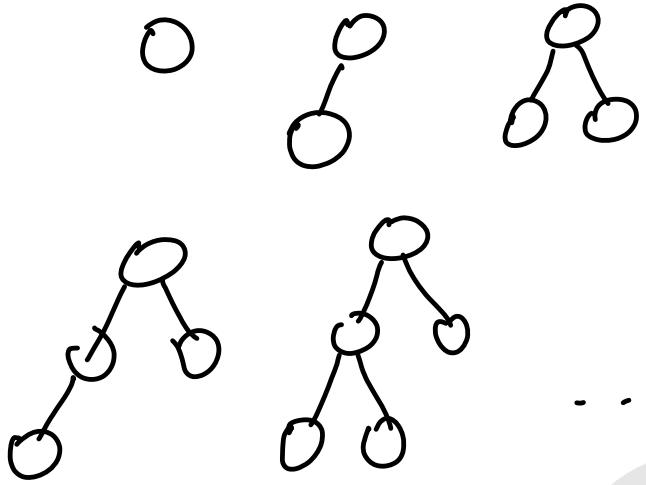
$D[7] = \cancel{\infty}$

$= 35$

Heap.



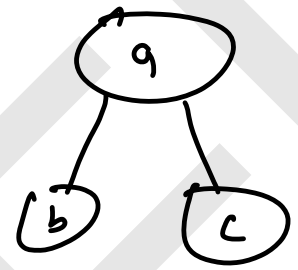




heap with n nodes will have
O(log n) levels. ←

Heap order property.

Min-Heap order.

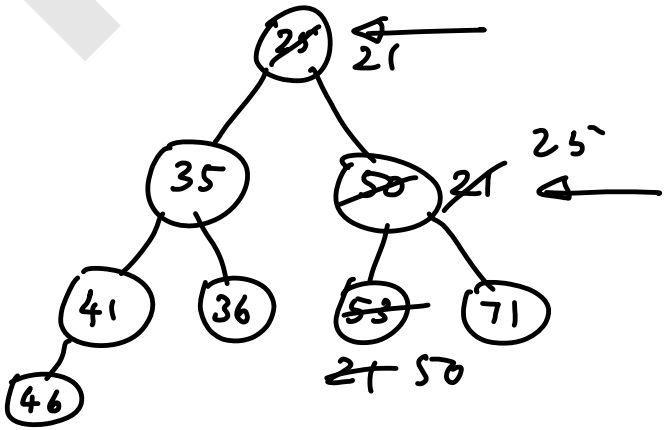


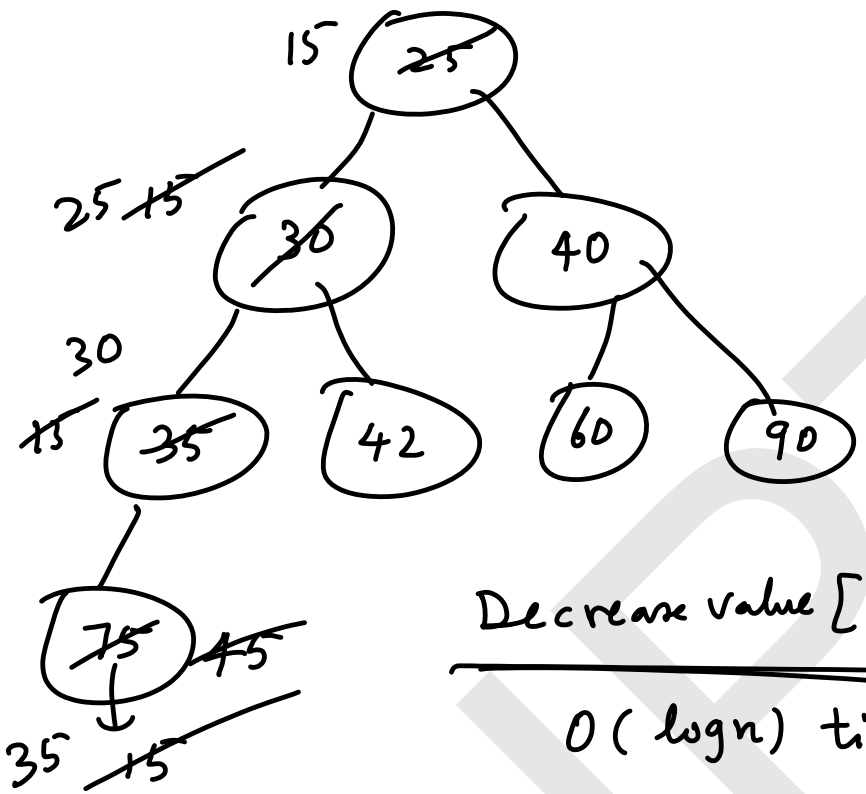
$a < b$,
$a < c$,.

Find Min

Delete Min

O(log n) time





Decrease value [D[v]]

$O(\log n)$ time.

$\{ D[v] \mid v \in V-S \}$

Heap

$\log n$ — Delete Min

Total cost for moving to S } $O(n \log n)$

Total cost of updates } $m \log n$

Total complexity is

$O(n \log n + m \log n)$.

$\downarrow \quad \updownarrow$

$O(n^2 + m)$??

$m = O(n)$ Sparse graph.

$\downarrow$

$O(n^2 + n)$

$O(n^2)$

$\downarrow$

$O(n \log n + m \log n)$

$O(n \log n)$

For Sparse graphs Heap based impl. is faster.

For Dense graph.

$m = O(n^2)$

$O(n^2 + m)$

$O(n^2 + n^2)$

$O(n^2)$

$O(n \log n + m \log n)$

$O(n \log n + n^2 \log n)$

$O(n^2 \log n)$

For Dense graph

Array Based Imp is

Faster than Heap based Imp.

Fibonacci Heap.

---

m Decrease value operations will take only $O(m)$ time.

The complexity is

$O(n \log n + m)$

$O(n^2 + m)$, $O(n \log n + m \log n)$

Dijkstra Algo Works only if $w(e) \geq 0$
