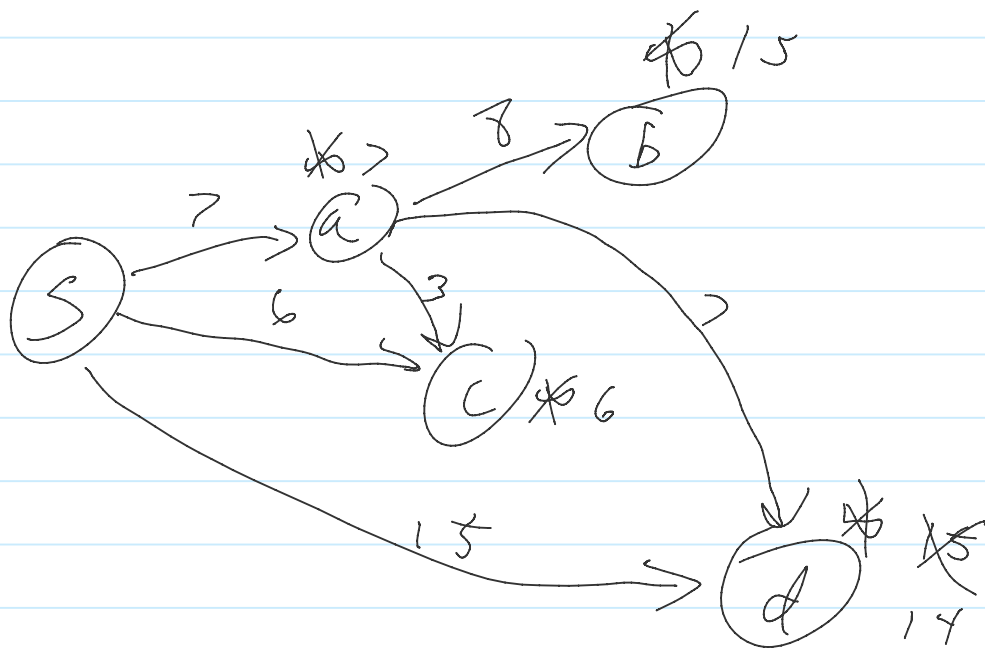


3/5 Dijkstra's

Monday, March 5, 2018 6:19 PM

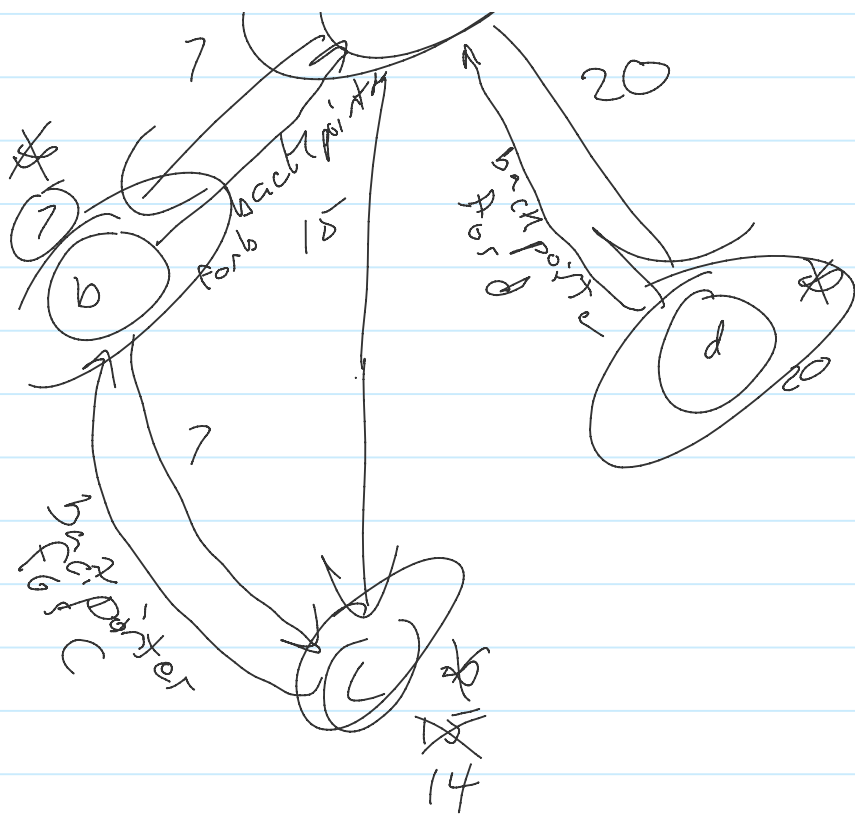
1. Initialize
visited array everything unvisited
distances set to ∞
starting distance is 0
current is starting node
2. Traversing edges
update distance based on current node
and its adjacent nodes.
3. Mark current node
visited
4. break out if there are no unvisited nodes with
finite distance or end node is visited
5. loop
current node = smallest distance unvisited node
goto step 2



Non negative edge weights for
Dijkstra's Algorithm to work

Back pointers can be used to extract optimal paths.





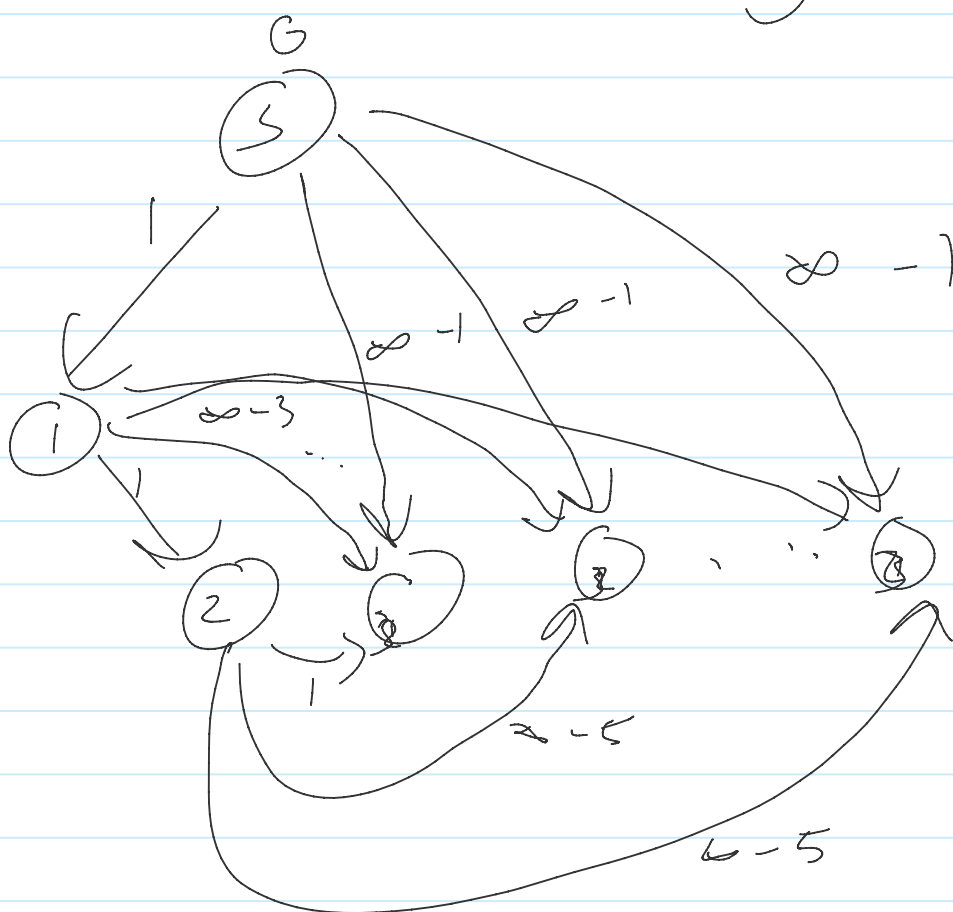
You should only use 1 back pointer per node, but depending on the reason for using Dijkstra's you might want more.

Big O Runtime $n = |V|$ $m = |E|$

BFS $O(|E| + |V|)$

Dijkstra's $O(n \log(n) + m)$ ← not quite

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



$$1 + 2 + 3 + 4 + 5 + \dots + n-1 \in O(n^2) = O(n)$$

assuming $n = |V|$ and
 $n = |E|$