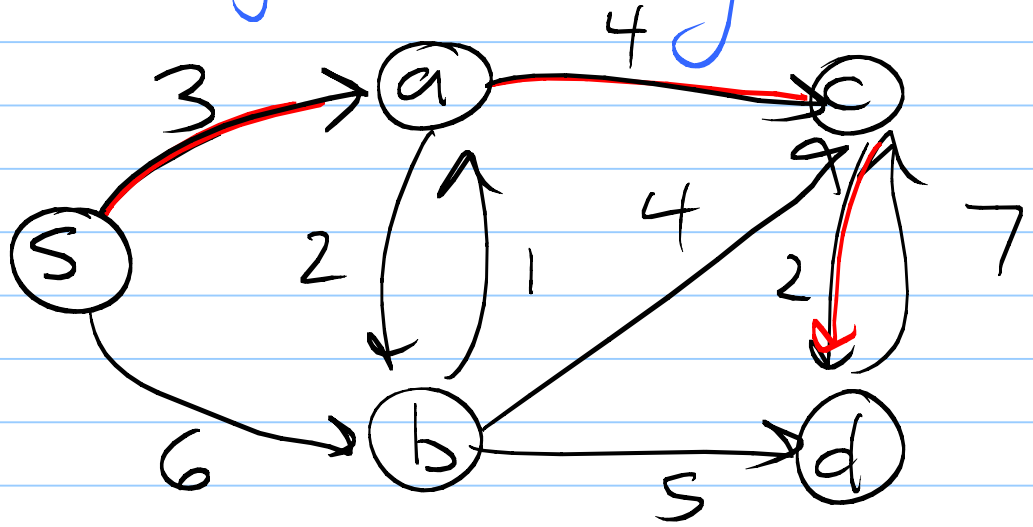
How can we solve the single source shortest path algorithm in a weighted graph where all weights ≥ 0?

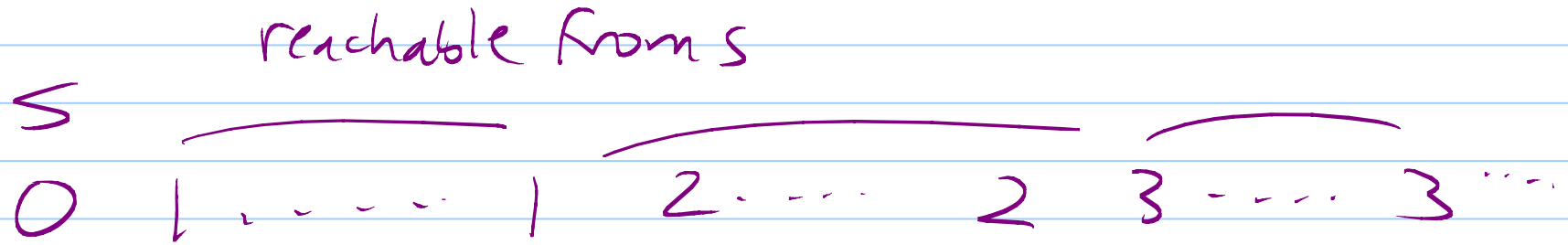
What goes wrong with bfs?



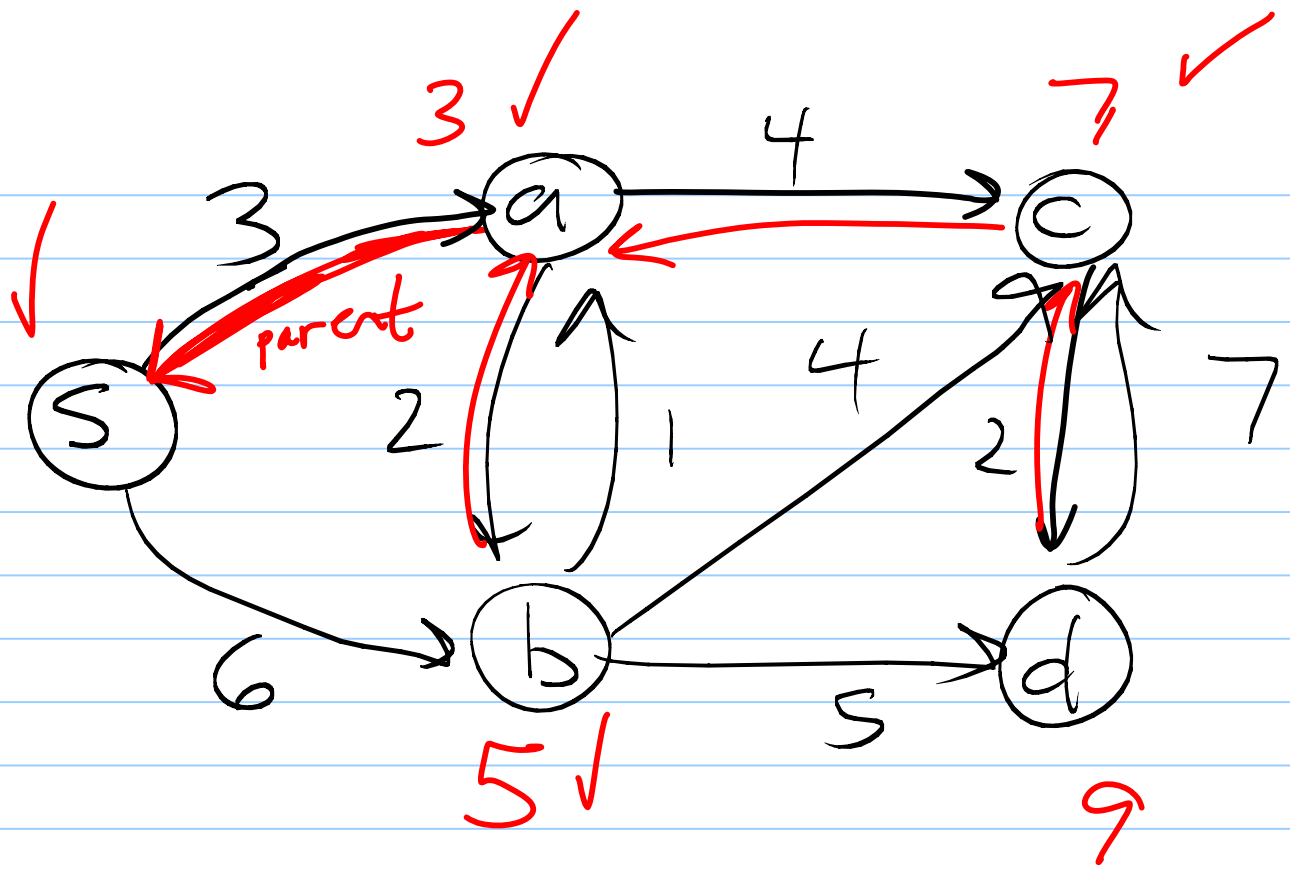
weight

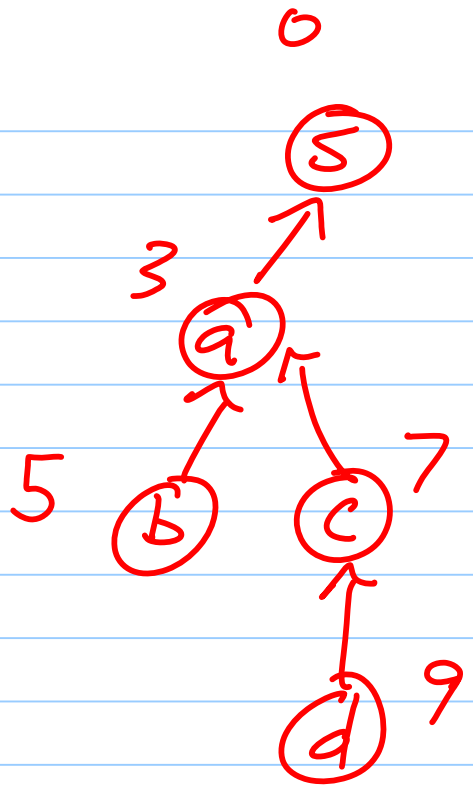
$3+4+2=9$

Use of queue in bfs was to process vertices in order of distance from source



occurs when removed from queue { Picked the reachable vertex in queue (not yet "placed in shortest path tree") with shortest distance from source


3
7
4
3
parent
s
a
c
2
1
4
2
7
6
b
d
5
5
9


0
s
3
a
5
b
c
7
d
9

```
DijkstraSingleSourceShortestPath(Vertex s)
  s.tracker = pq.addTracked(0, s)
  For all v ∈ V - {s}
    v.tracker = pq.addTracked(∞, s)
  For all v ∈ V
    v.parent = null
  while (!pq.isEmpty())
    tag = pq.minTag()
    if (tag == ∞) return   // remain vertices (in pq) are not reachable from s
    u = pq.extractMin()
    u.dist = tag
    for each edge from u
      v = u.dest
      if (v.tracker.inCollection() &&
          u.dist + e.weight < v's current tag)
        v.tracker.update(u.dist + e.weight)
        v.parent = e
```

tag: semantics length of shortest path from s found so far