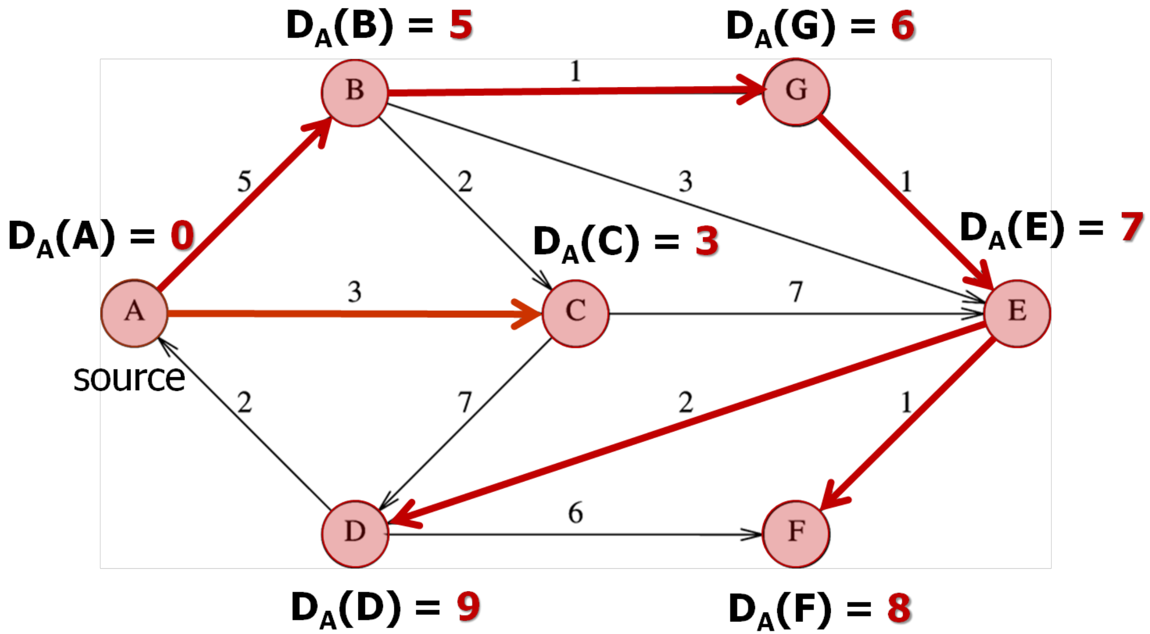
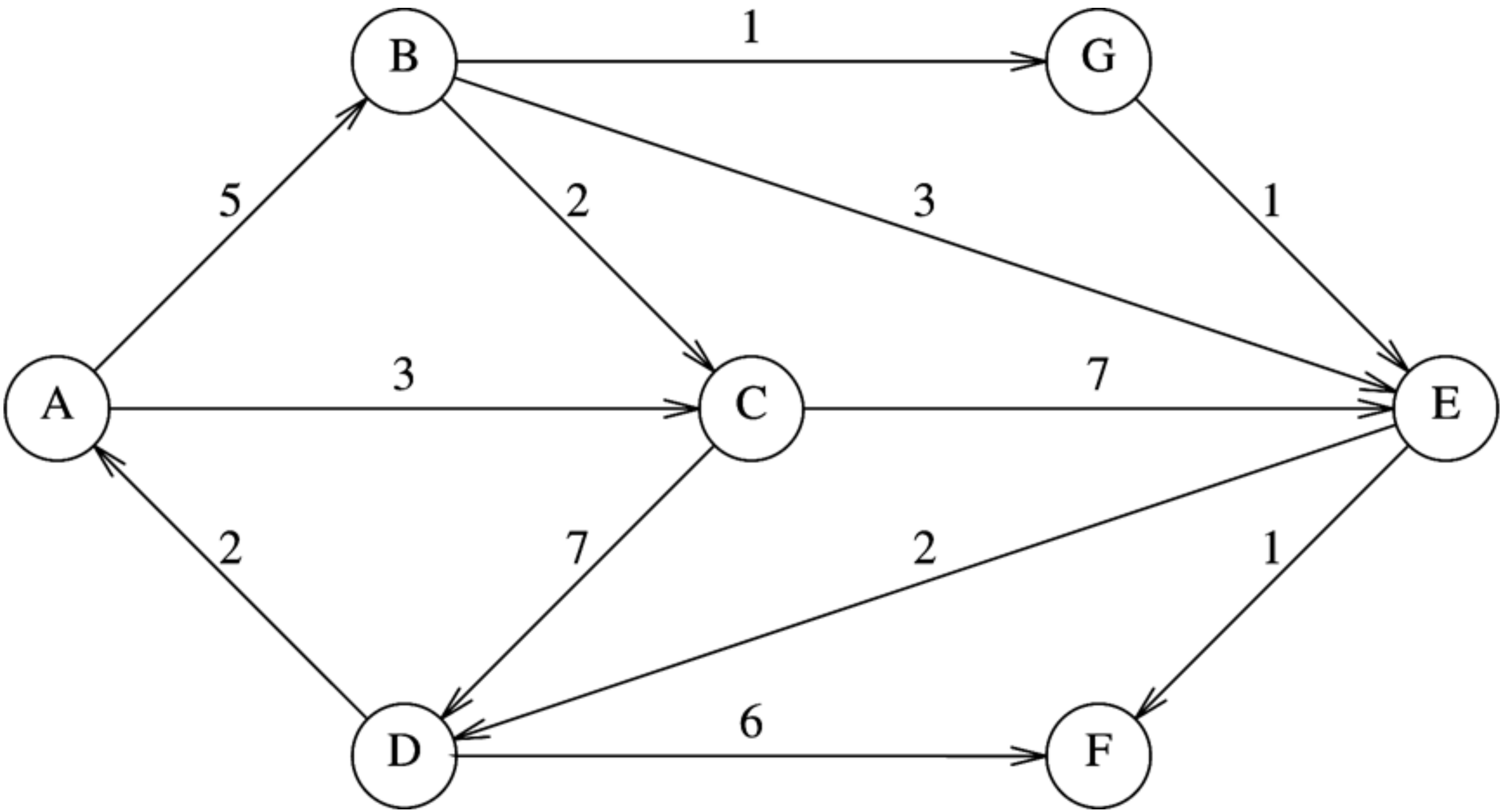
Dijkstra's Algorithm

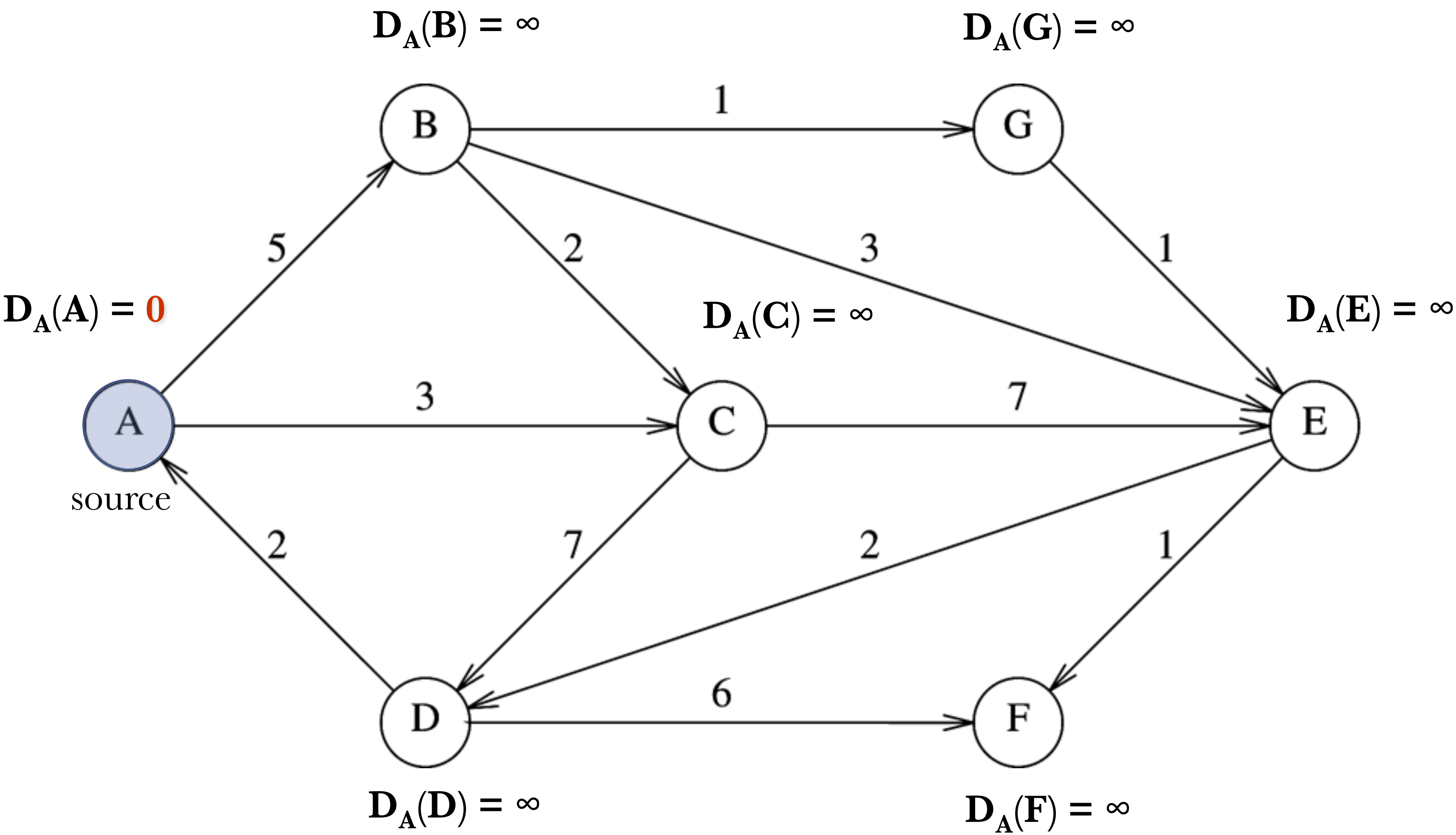


Directed Weighted Graph

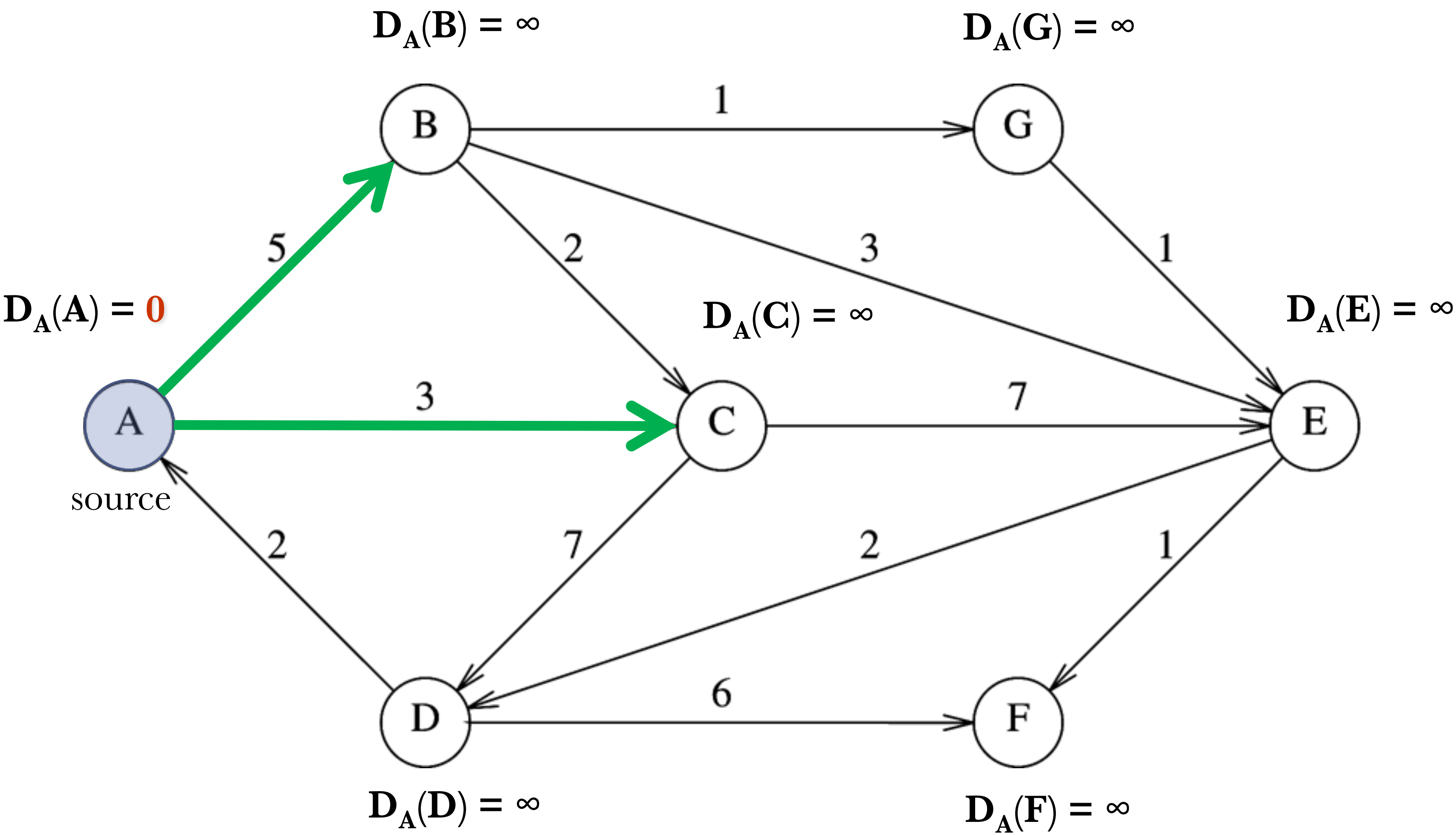


Algorithm Dijkstra

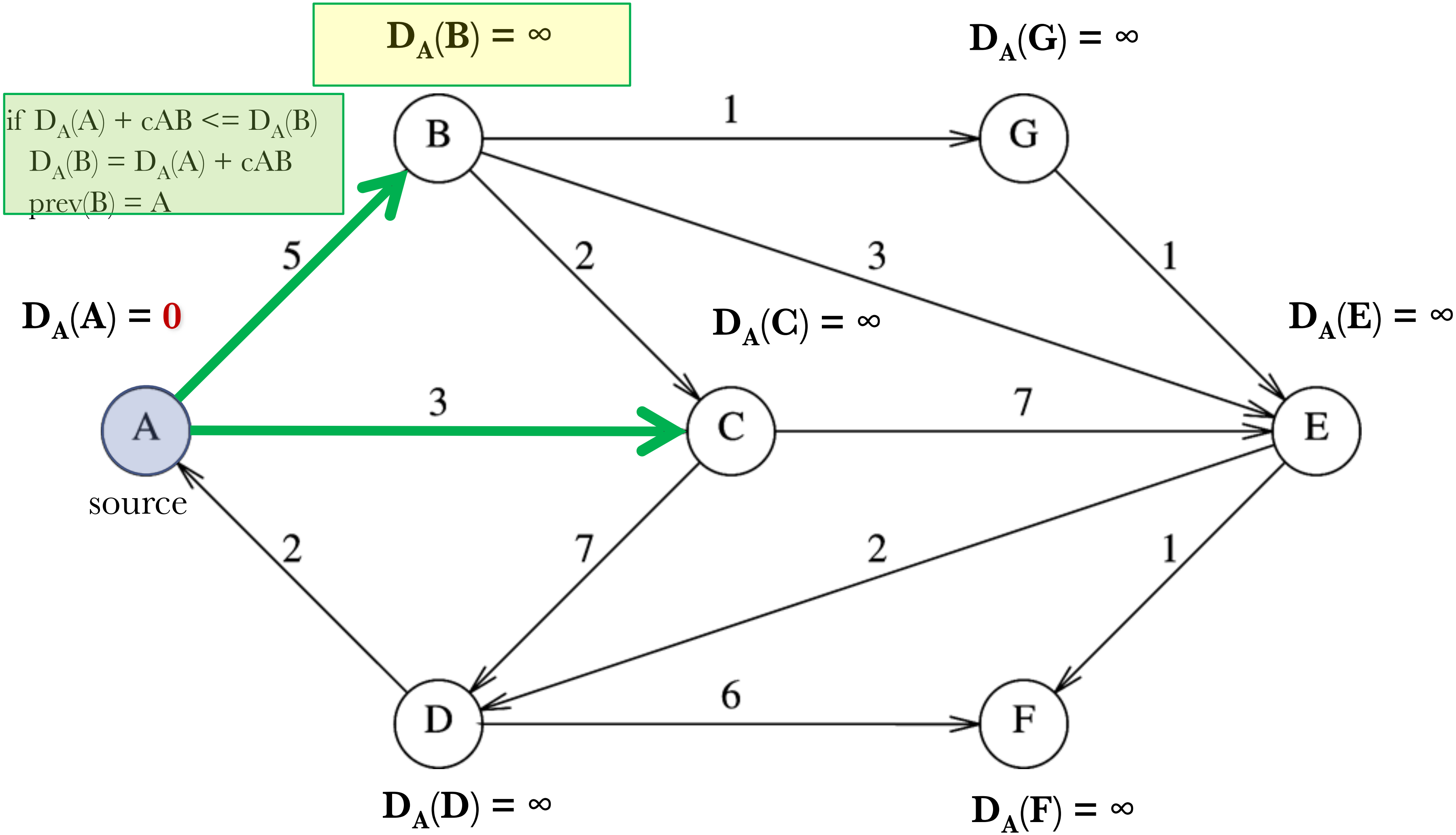
```
for each Vertex v do
  v.known <- false
  v.dist  <- infinity //distance from source vertex
  v.prev  <- NULL

  //s is the source vertex
   s.dist <- 0
   do n times
      v <- unknown vertex with minimum v.dist
      v.known <- true
      for each edge (v,w) do
          if v.dist + cvw <= w.dist then
              w.dist <- v.dist + cvw
              w.prev <- v
```



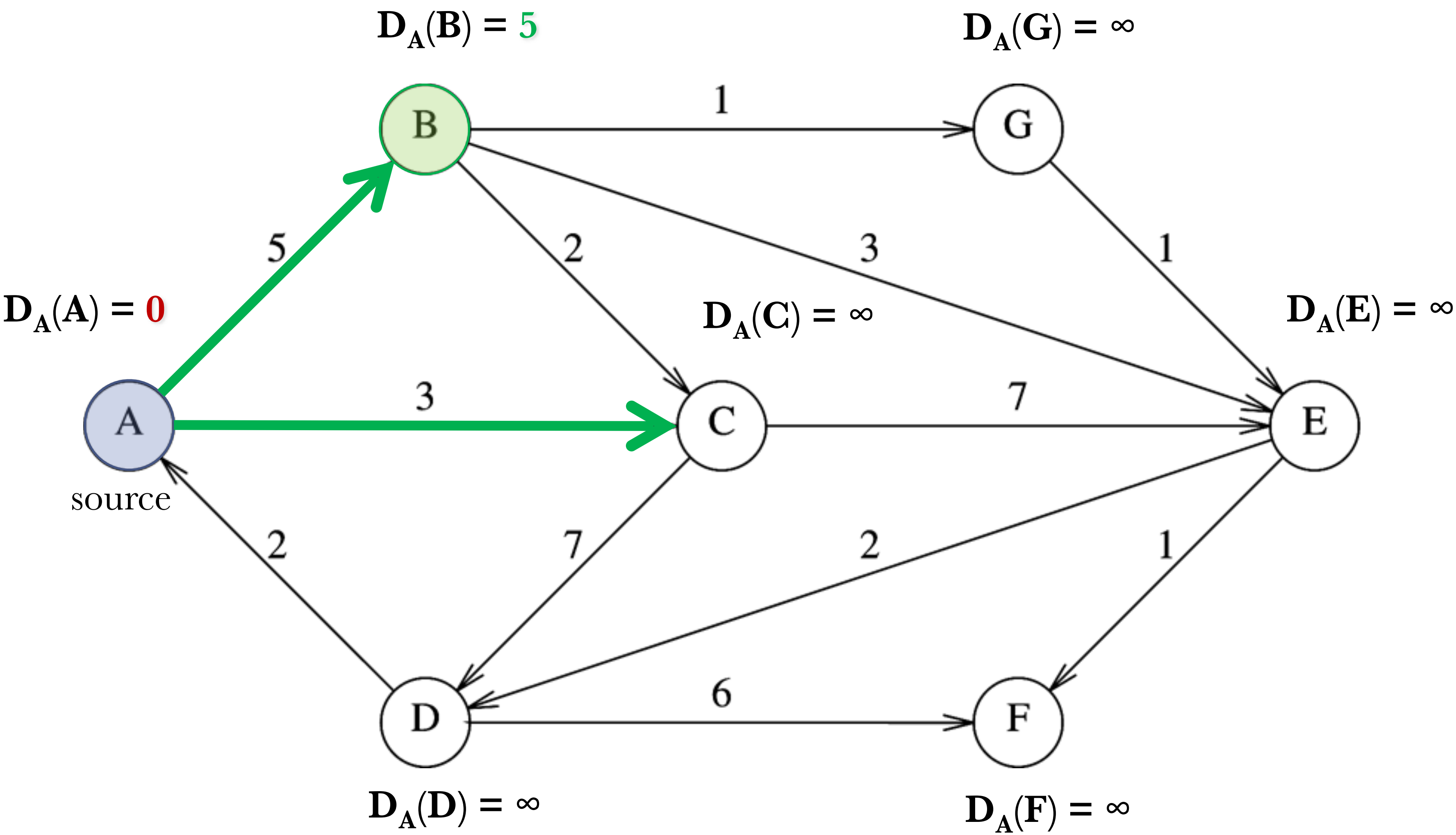
Weight $f(n) = D_{source}(Destination)$



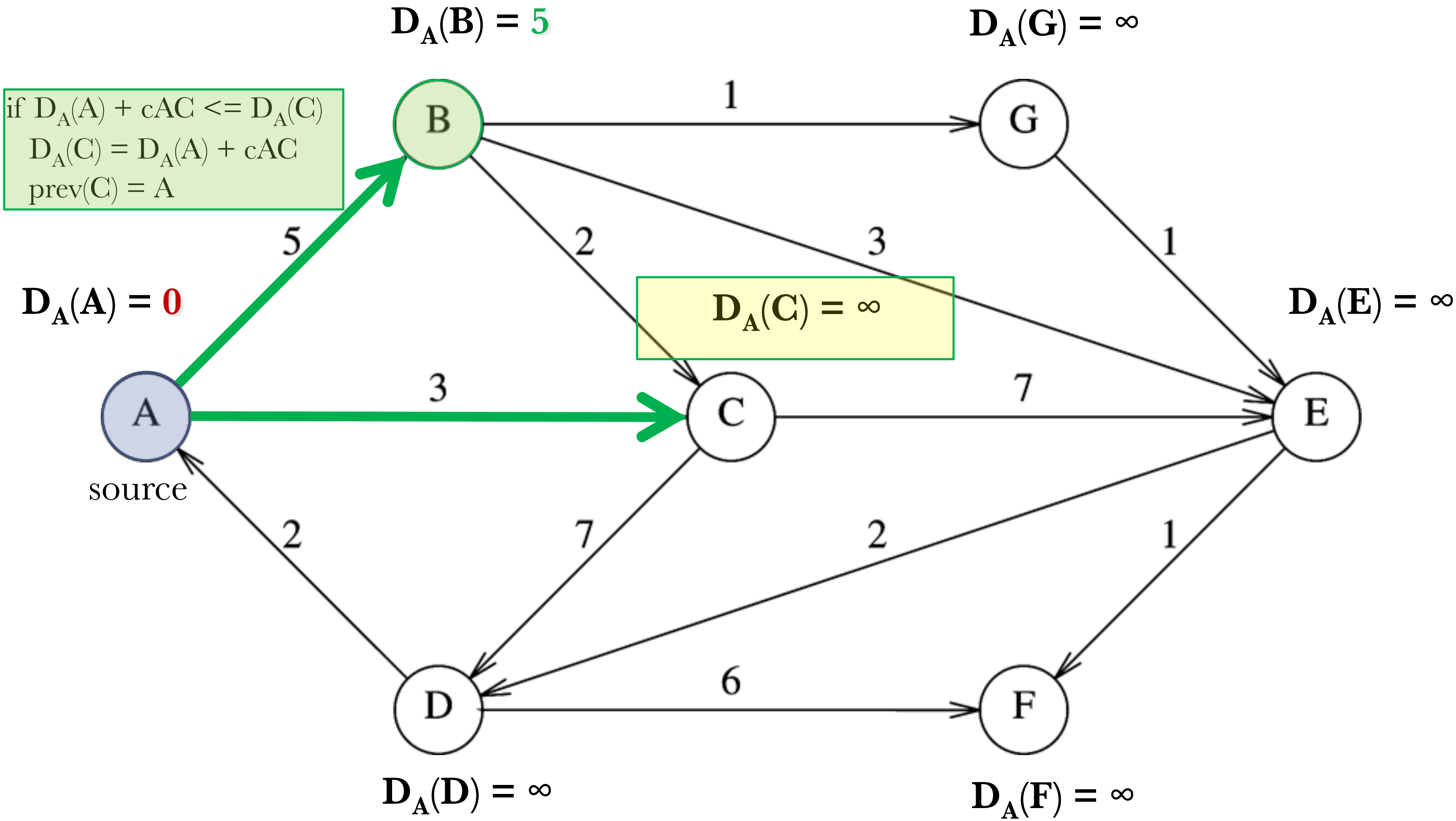
Weight f(n) = D_{source}(Destination)



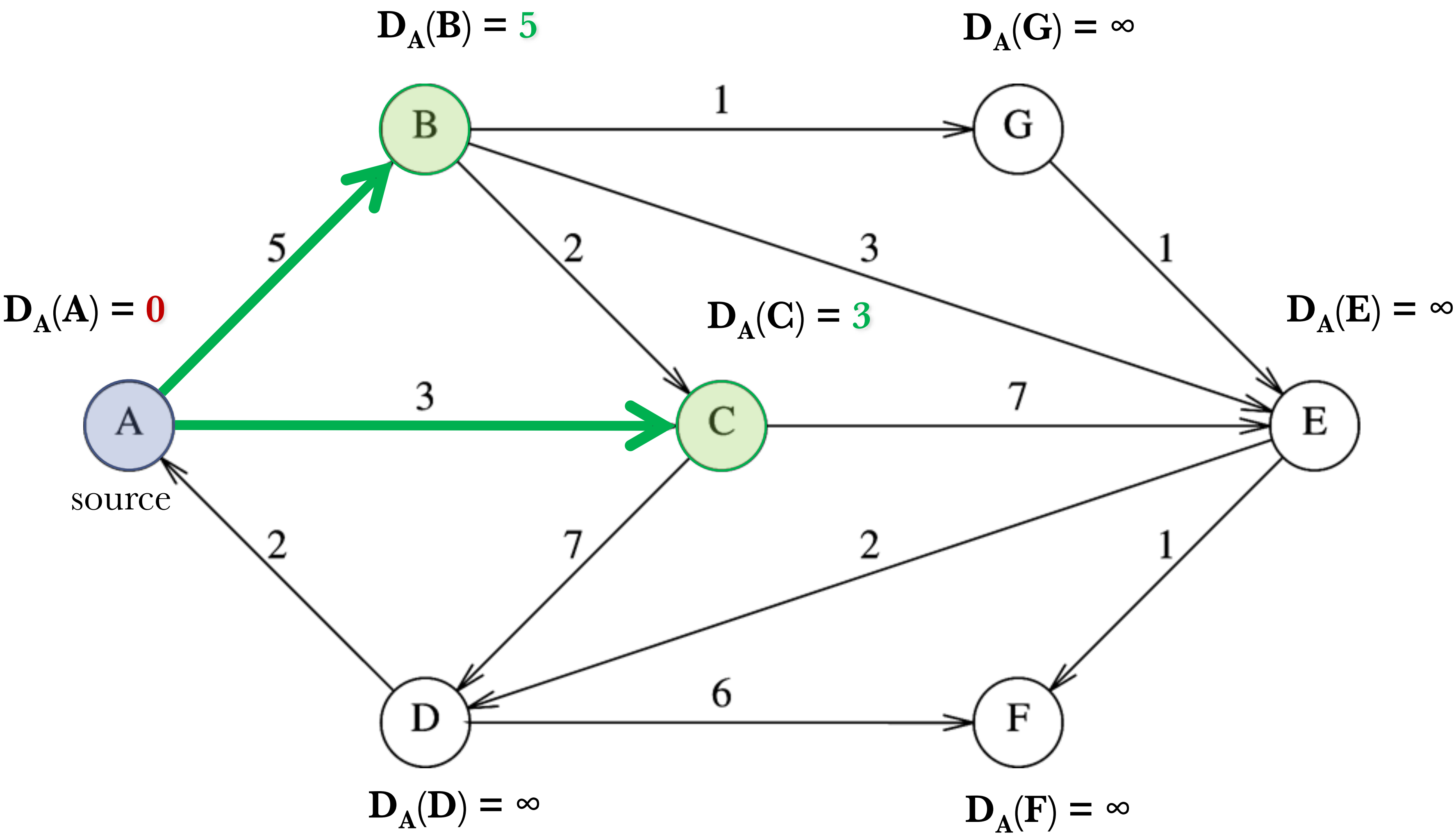
Weight f(n) = D_{source}(Destination)



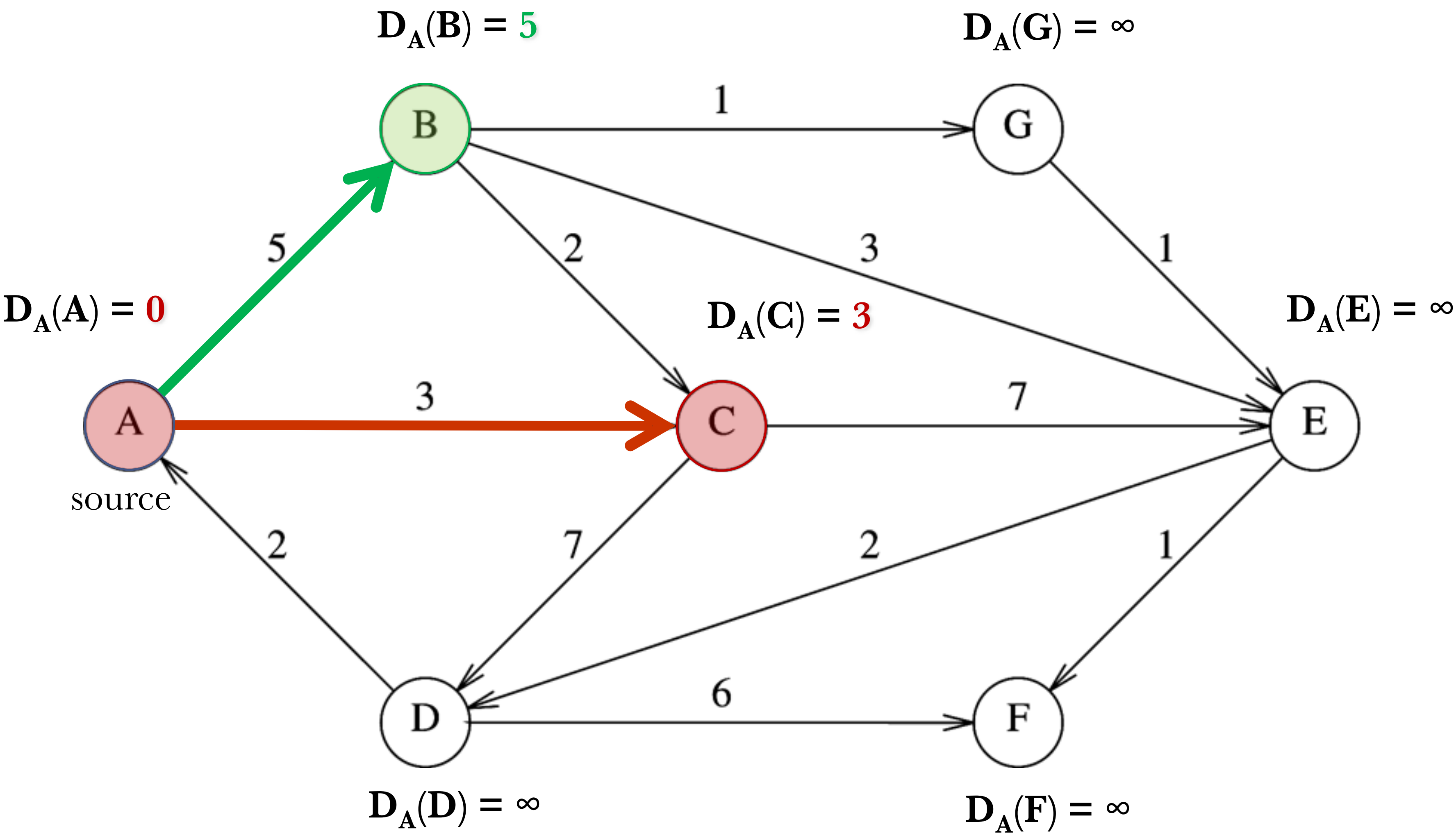
Weight $f(n) = D_{source}(Destination)$



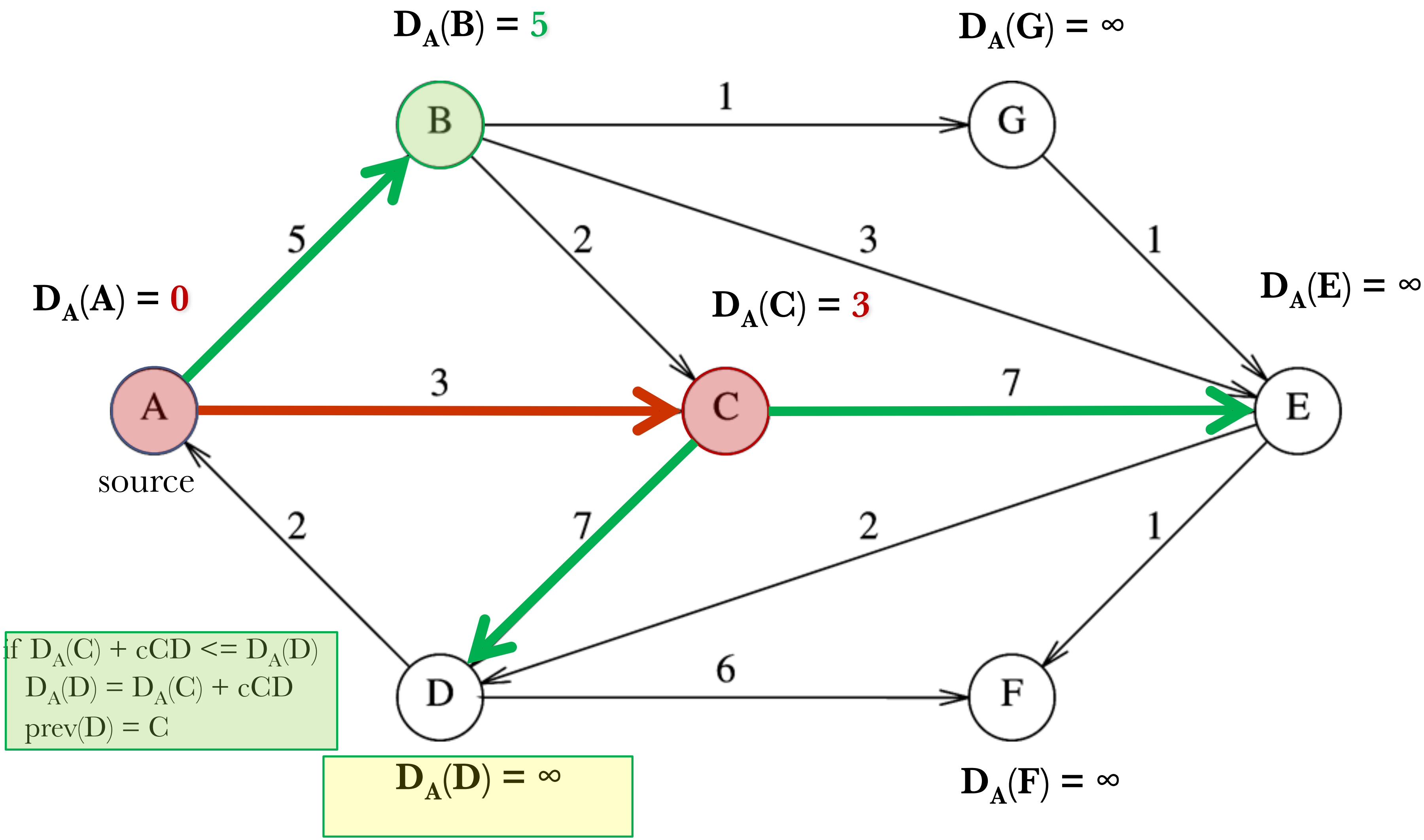
Weight $f(n) = D_{source}(Destination)$



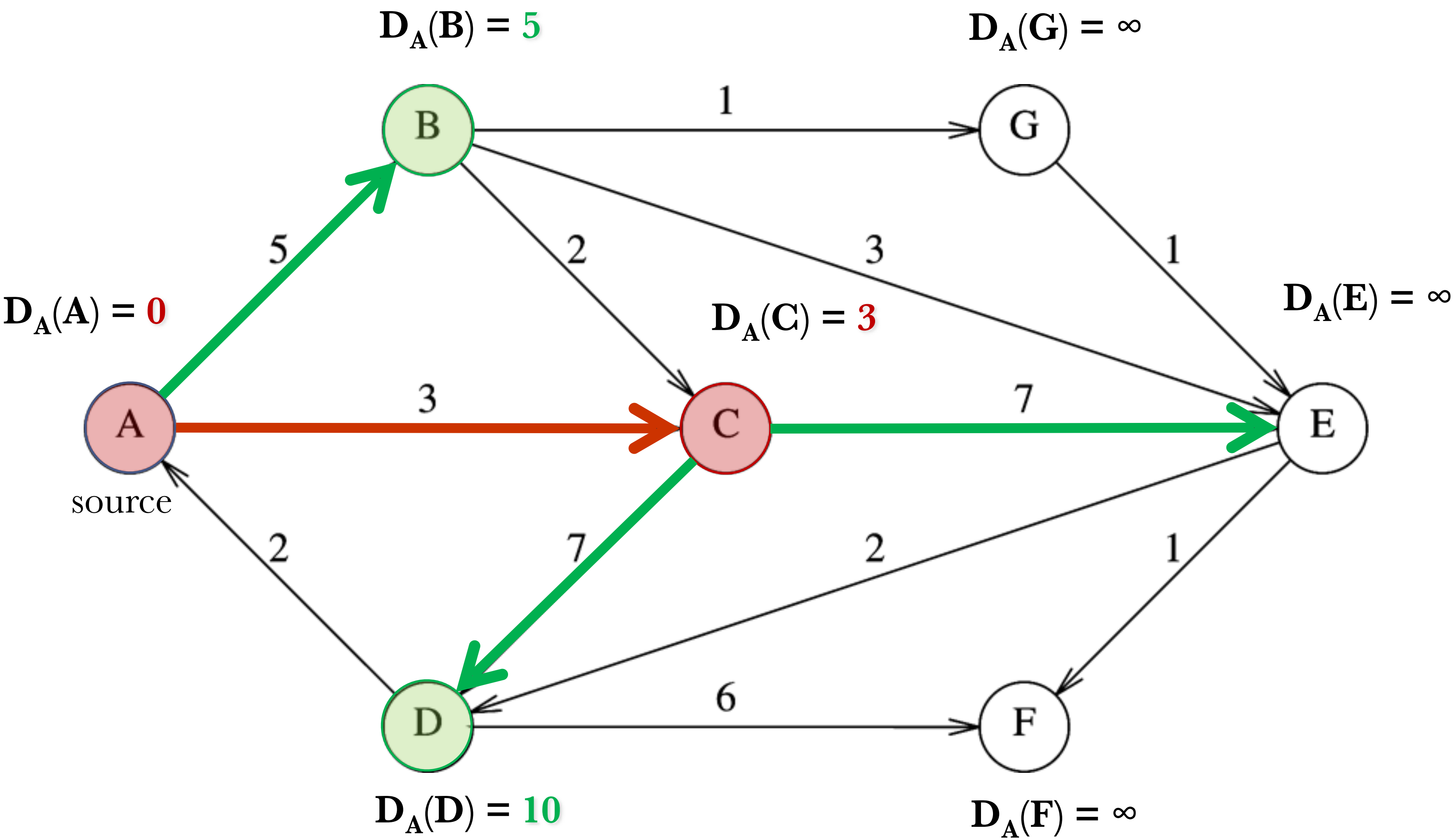
Weight $f(n) = D_{source}(Destination)$



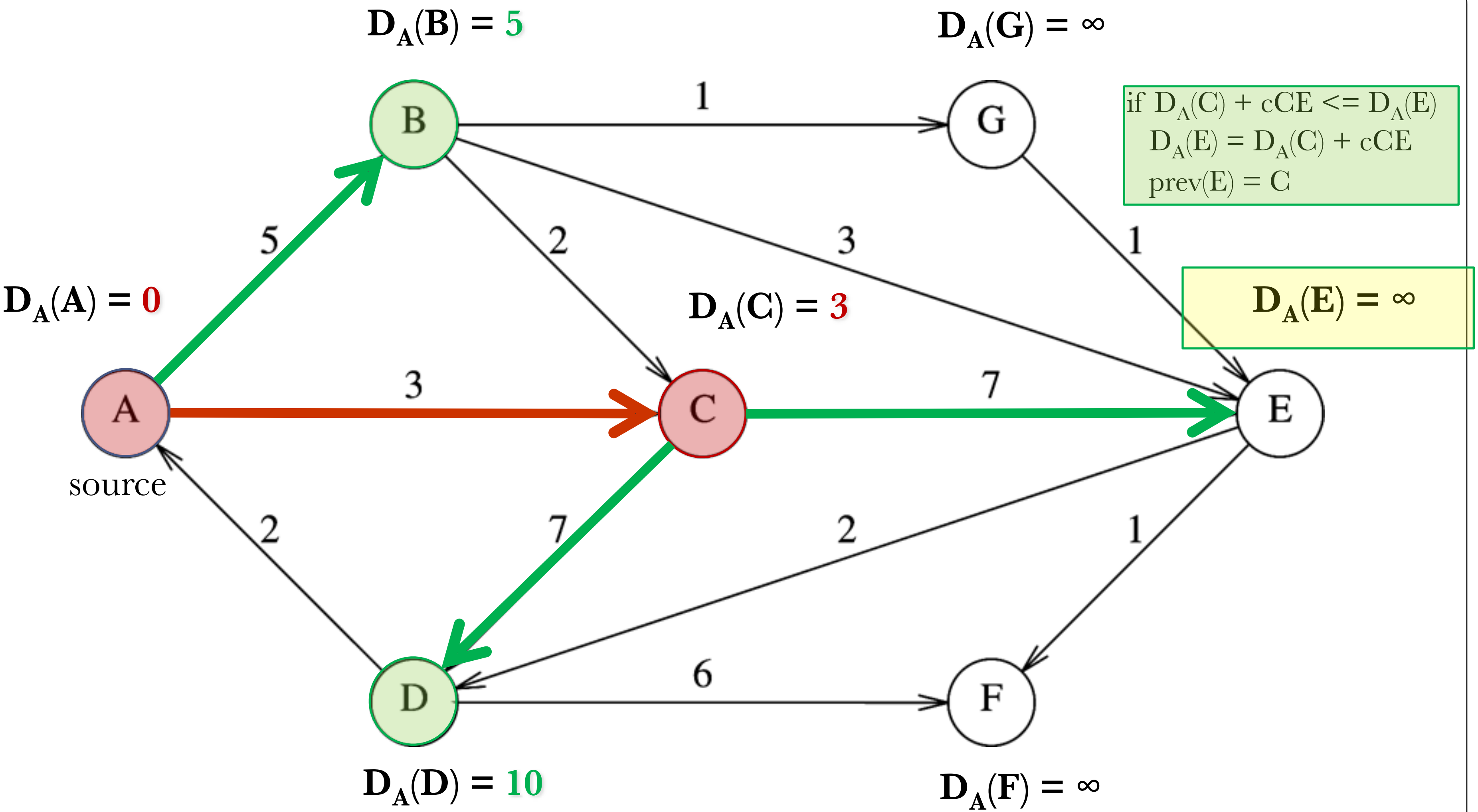
Weight f(n) = D_{source}(Destination)



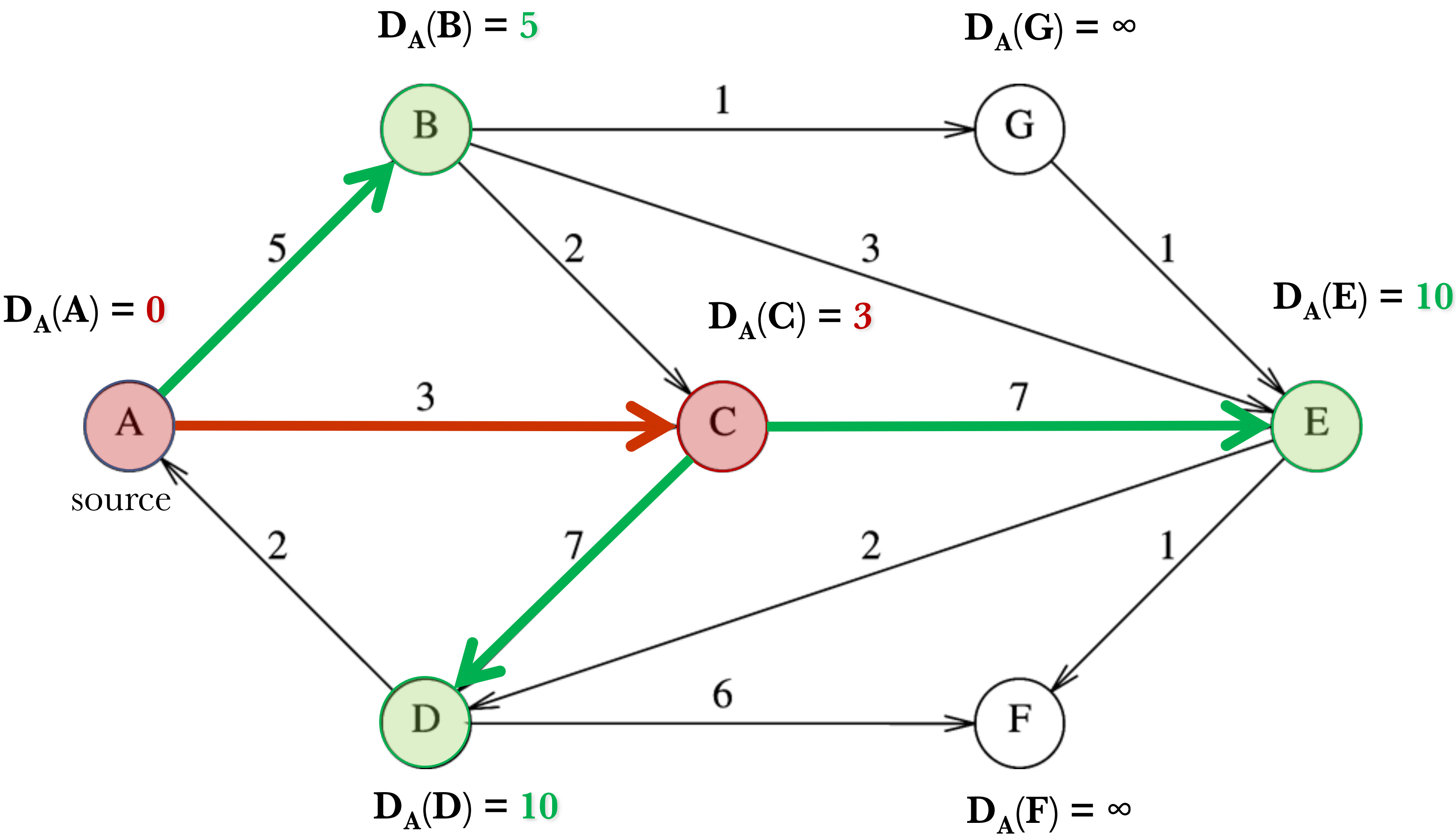
Weight f(n) = D_{source}(Destination)



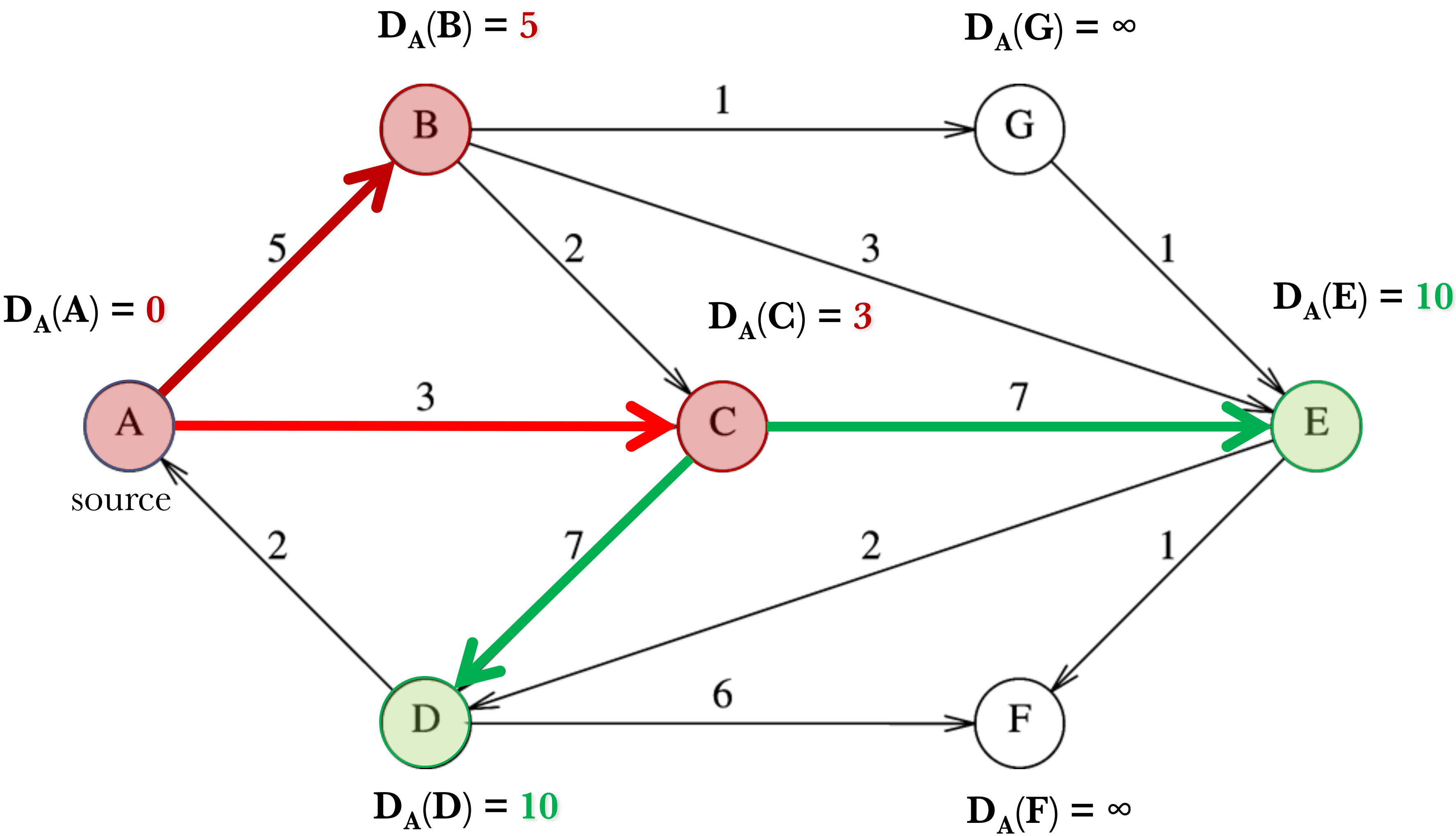
Weight f(n) = D_{source}(Destination)



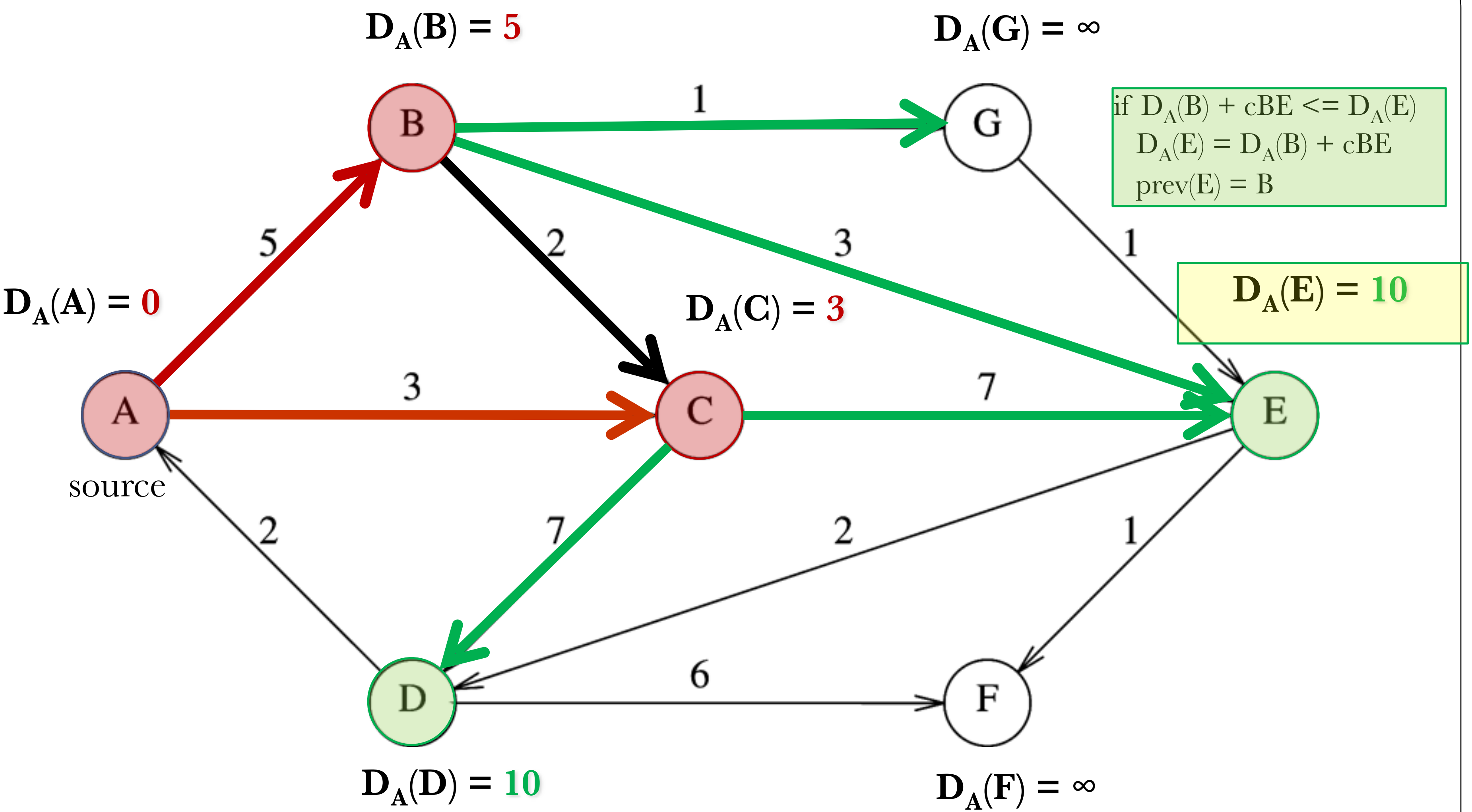
Weight f(n) = D_{source}(Destination)



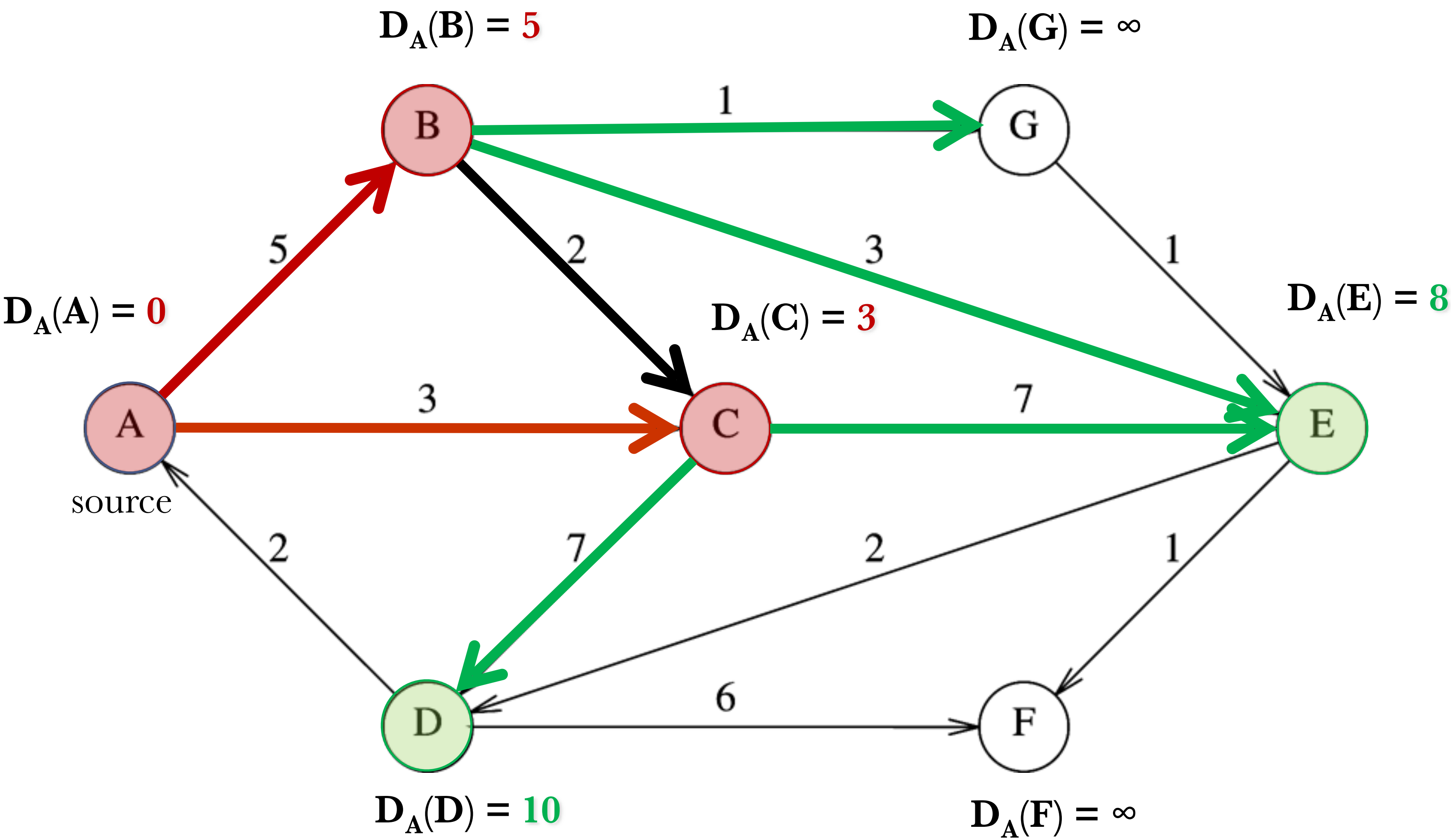
Weight $f(n) = D_{source}(Destination)$



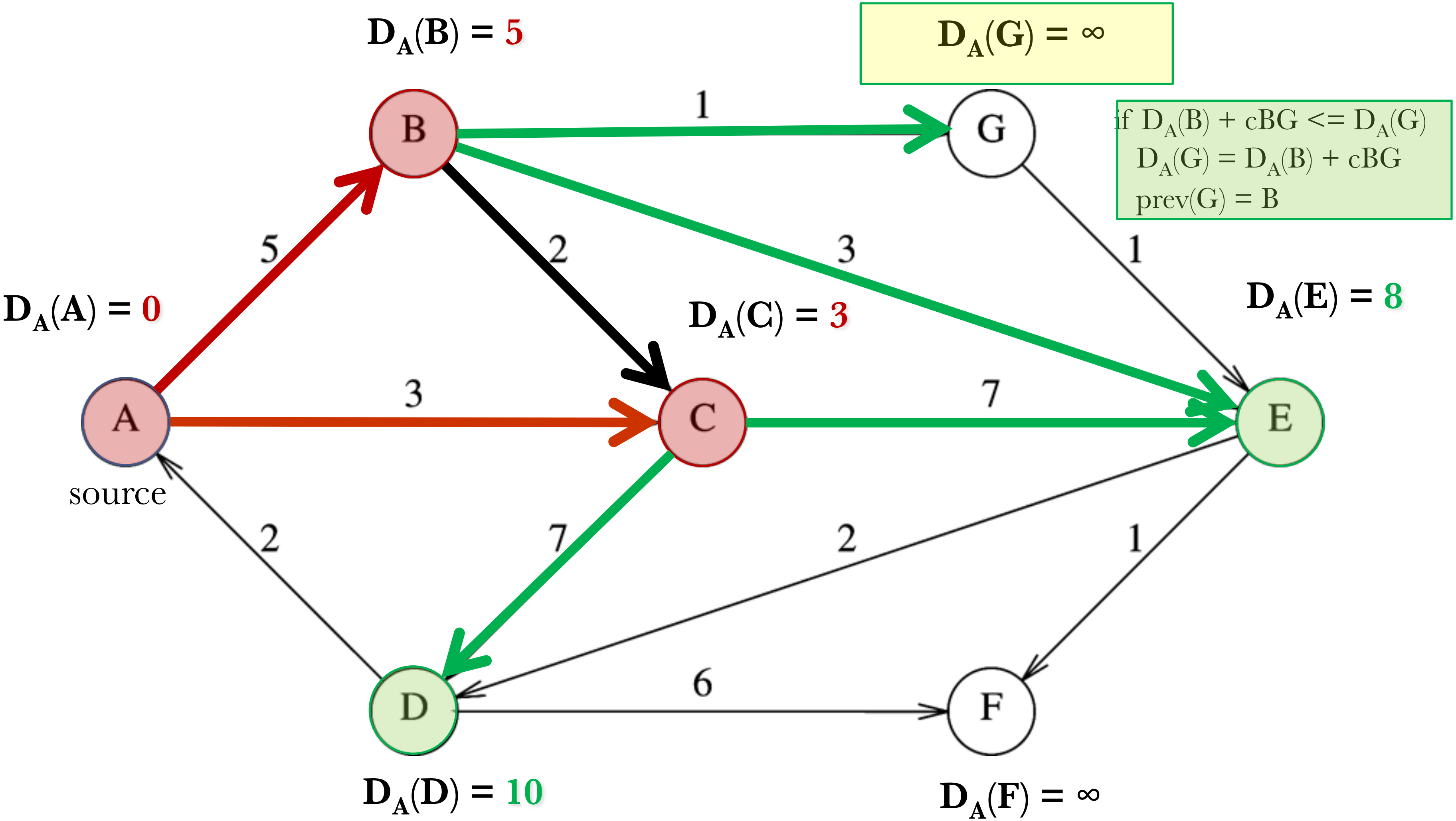
Weight f(n) = D_{source}(Destination)



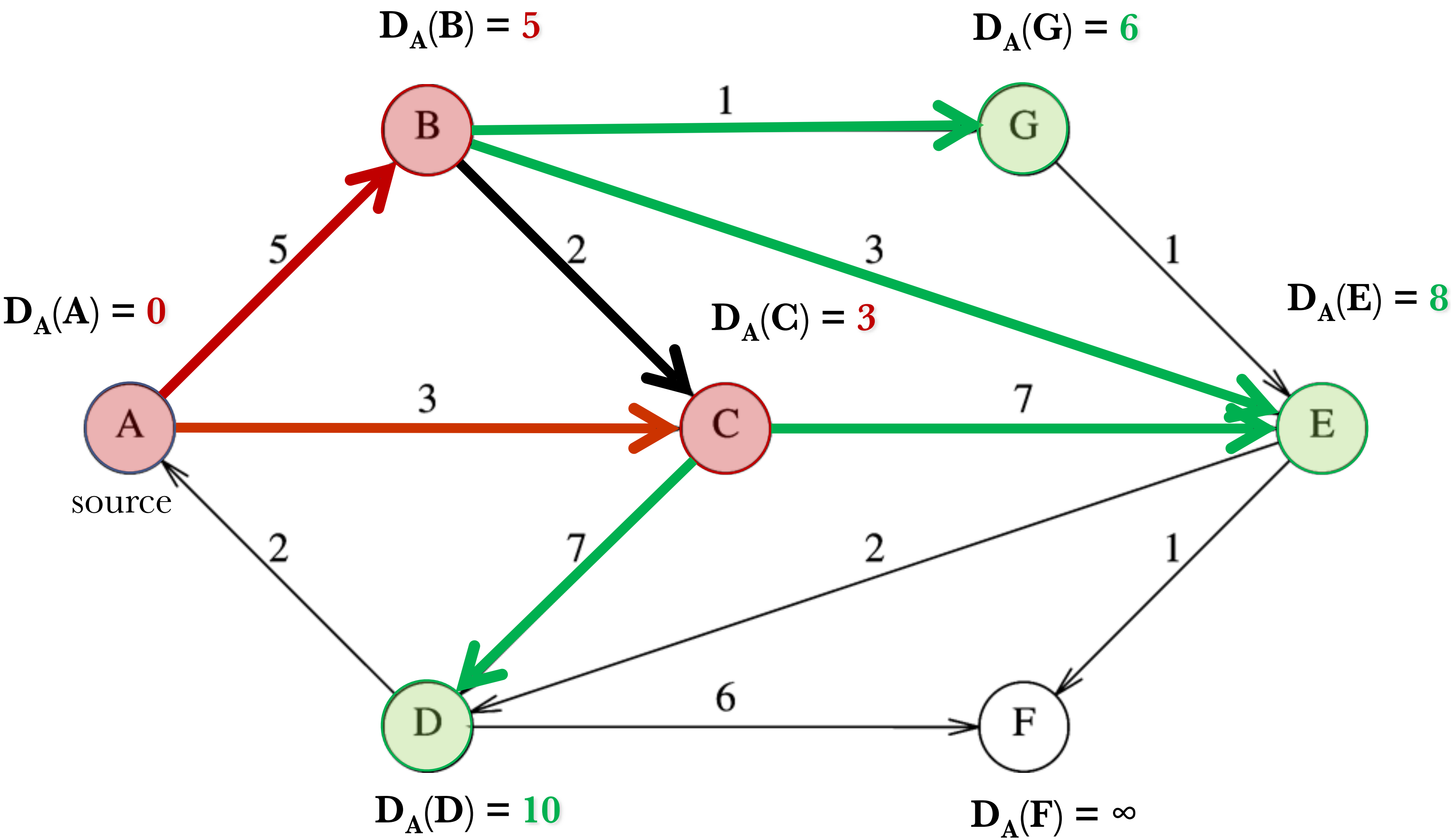
Weight f(n) = D_{source}(Destination)



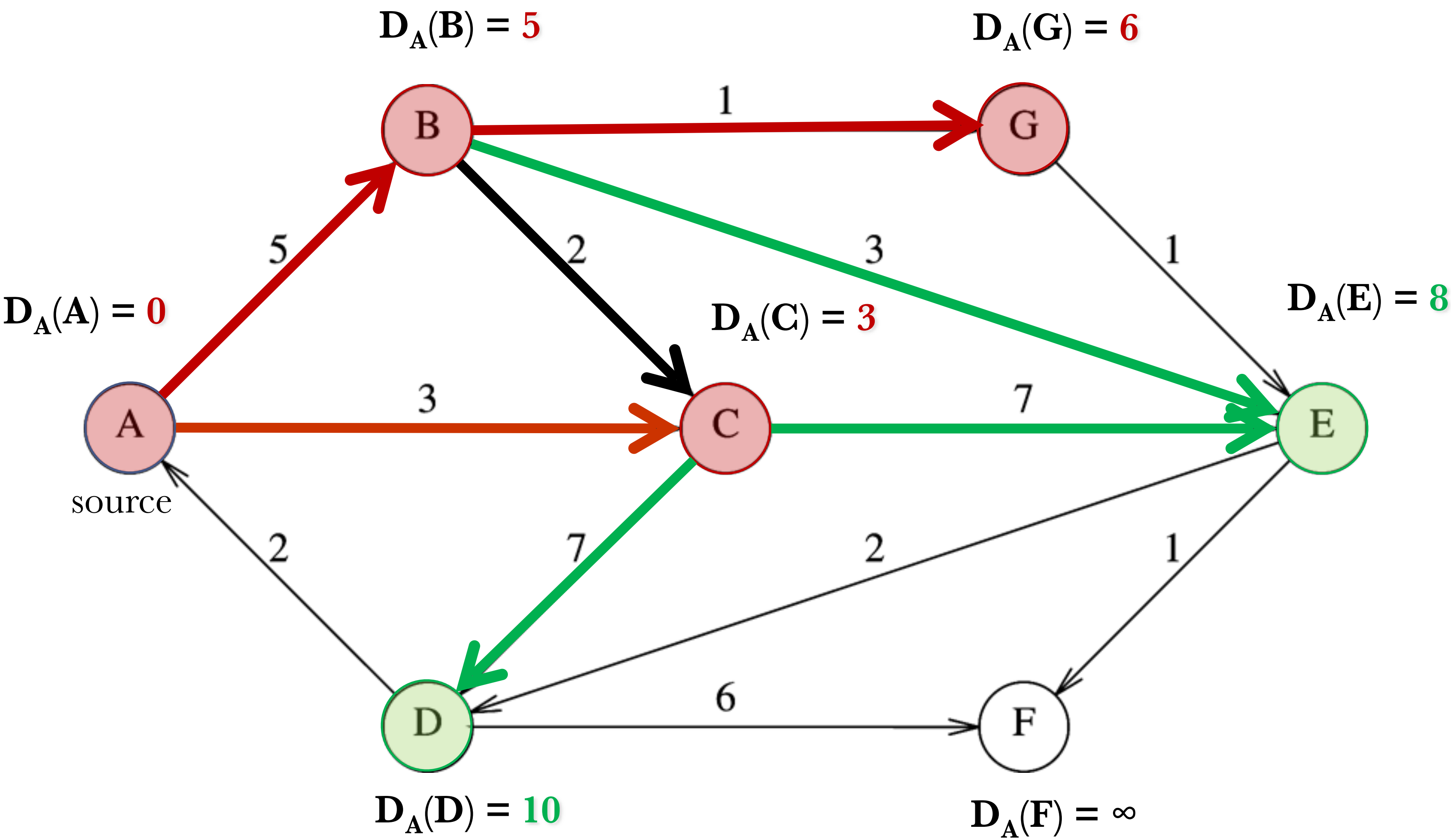
Weight $f(n) = D_{source}(Destination)$



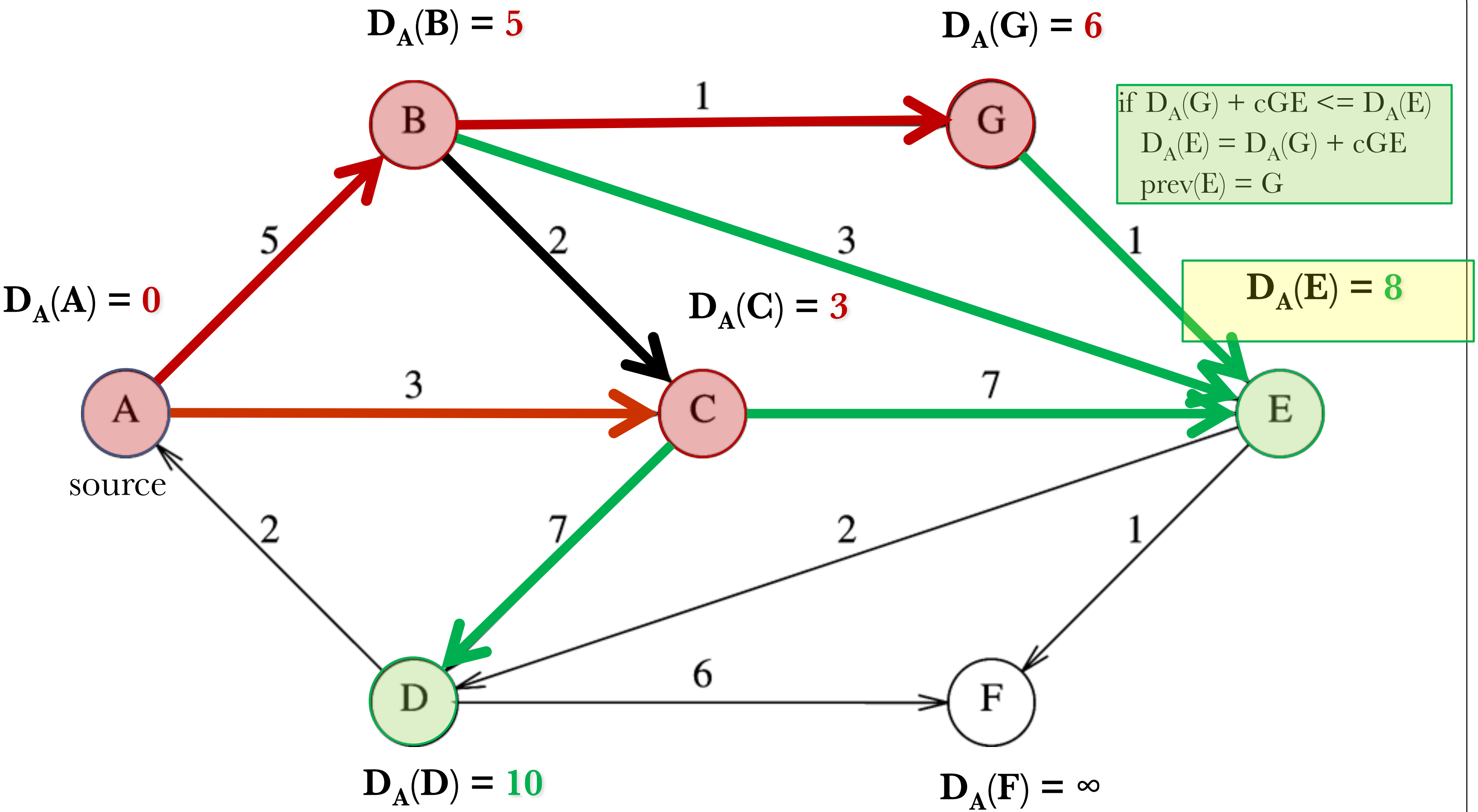
Weight $f(n) = D_{source}(Destination)$



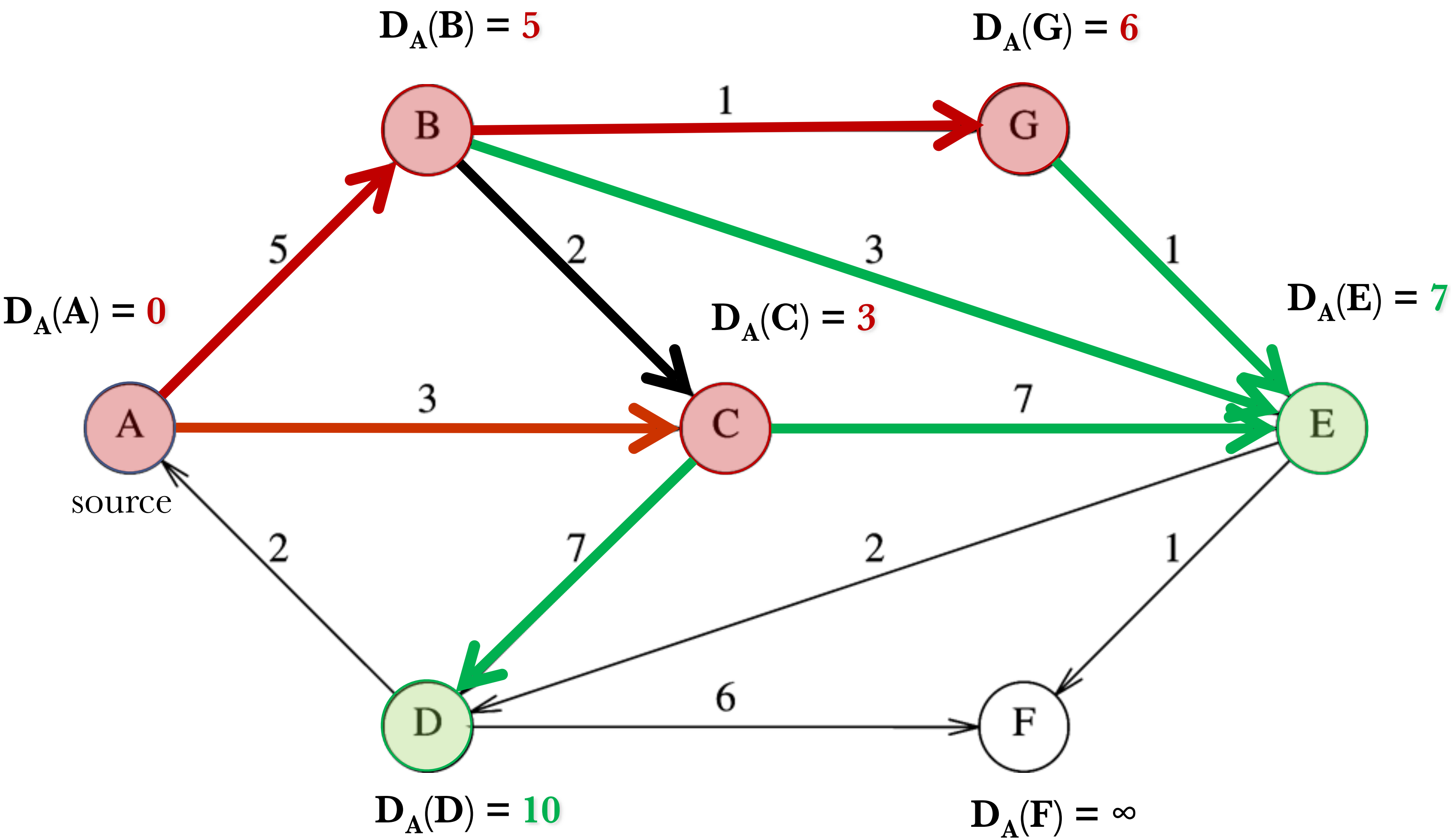
Weight f(n) = D_{source}(Destination)



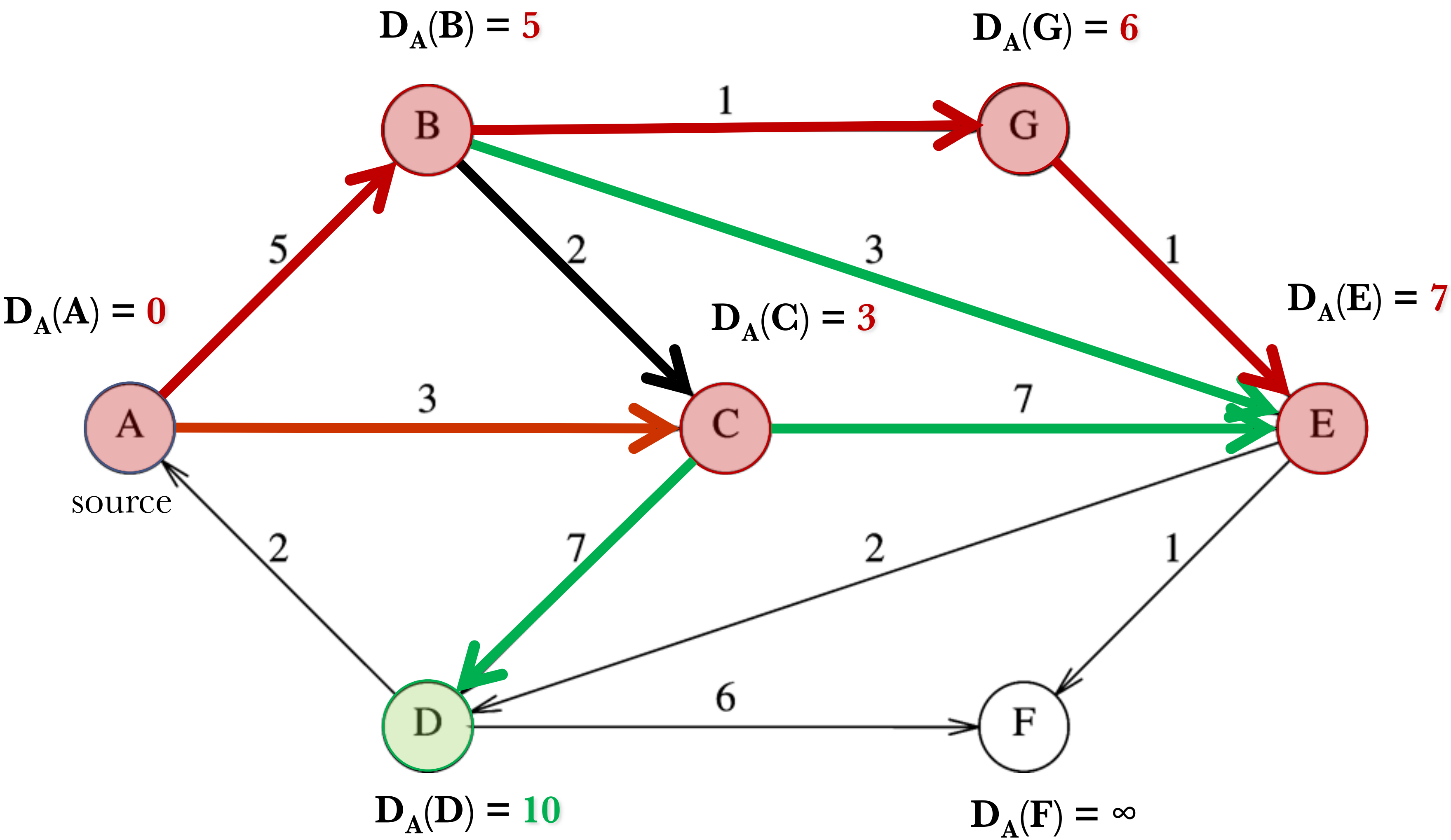
Weight f(n) = D_{source}(Destination)



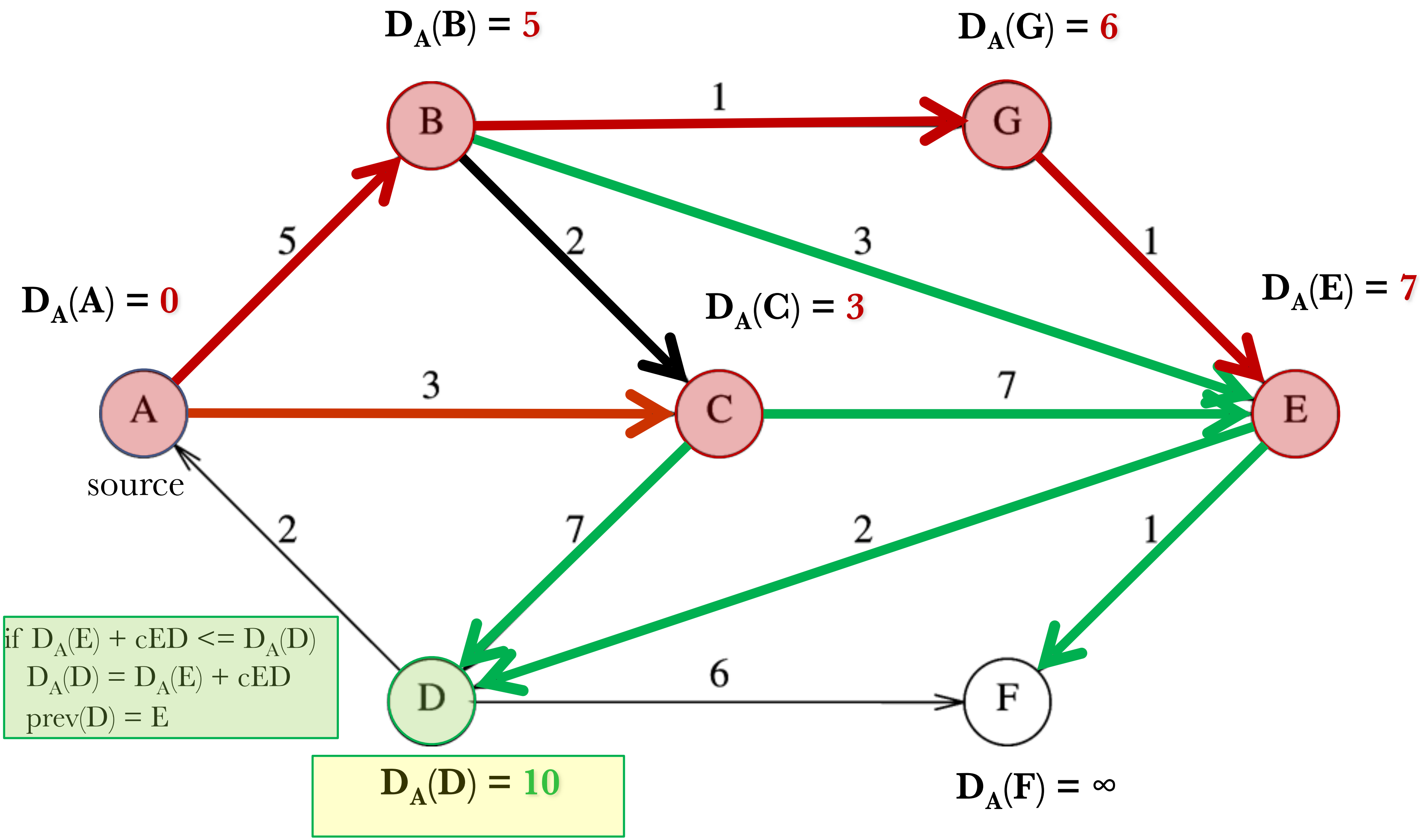
Weight f(n) = D_{source}(Destination)



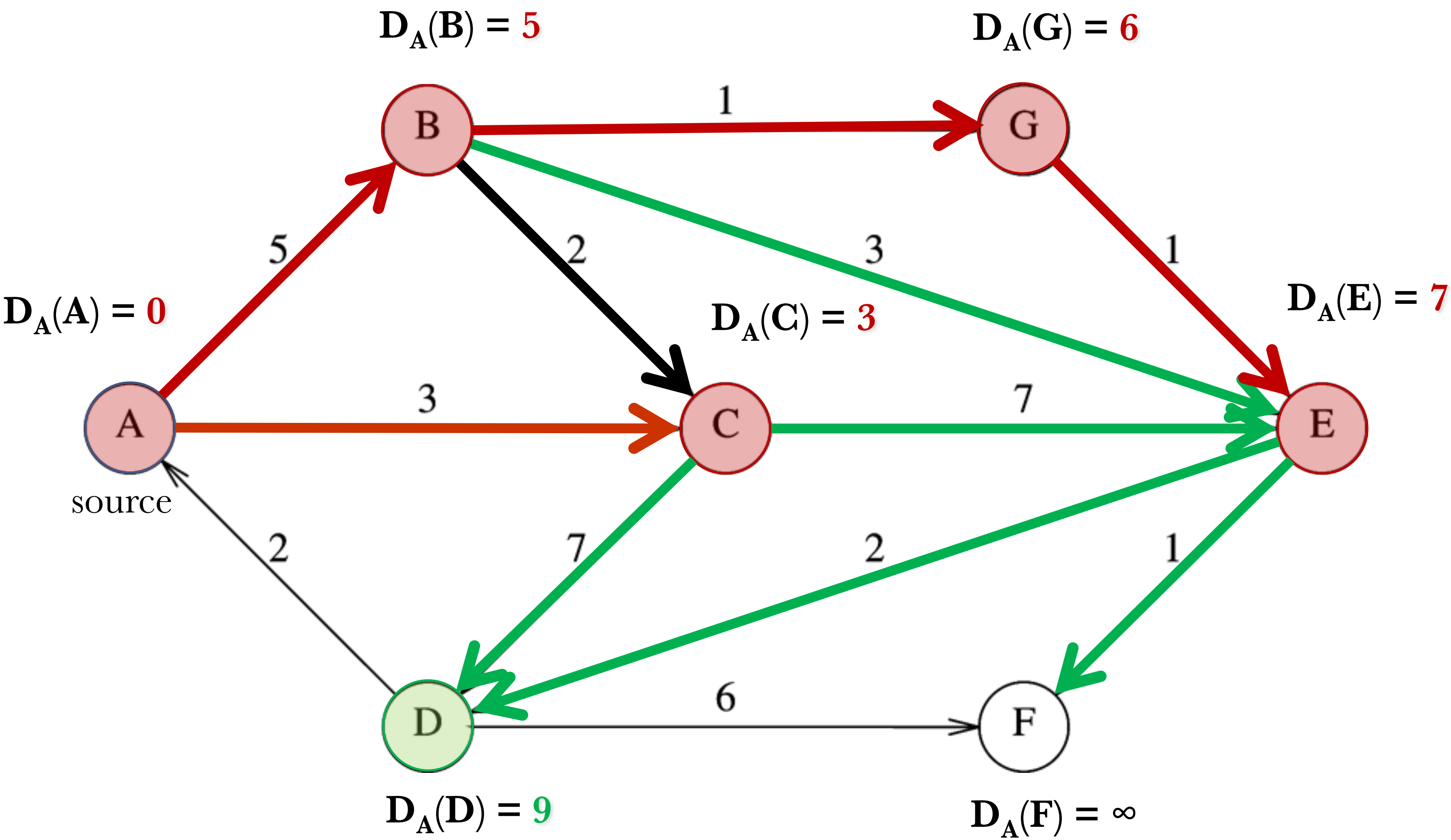
Weight $f(n) = D_{source}(Destination)$



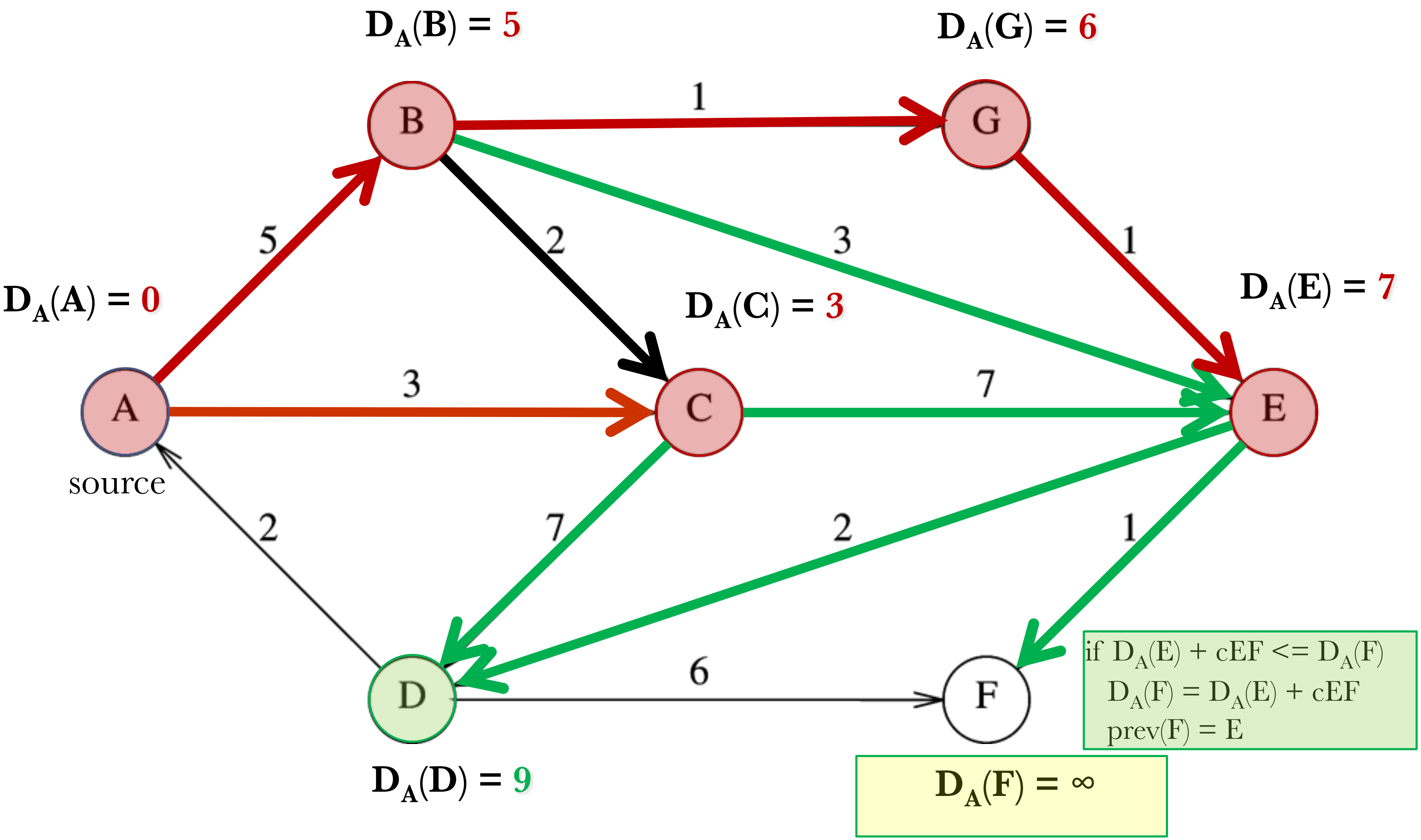
Weight f(n) = D_{source}(Destination)



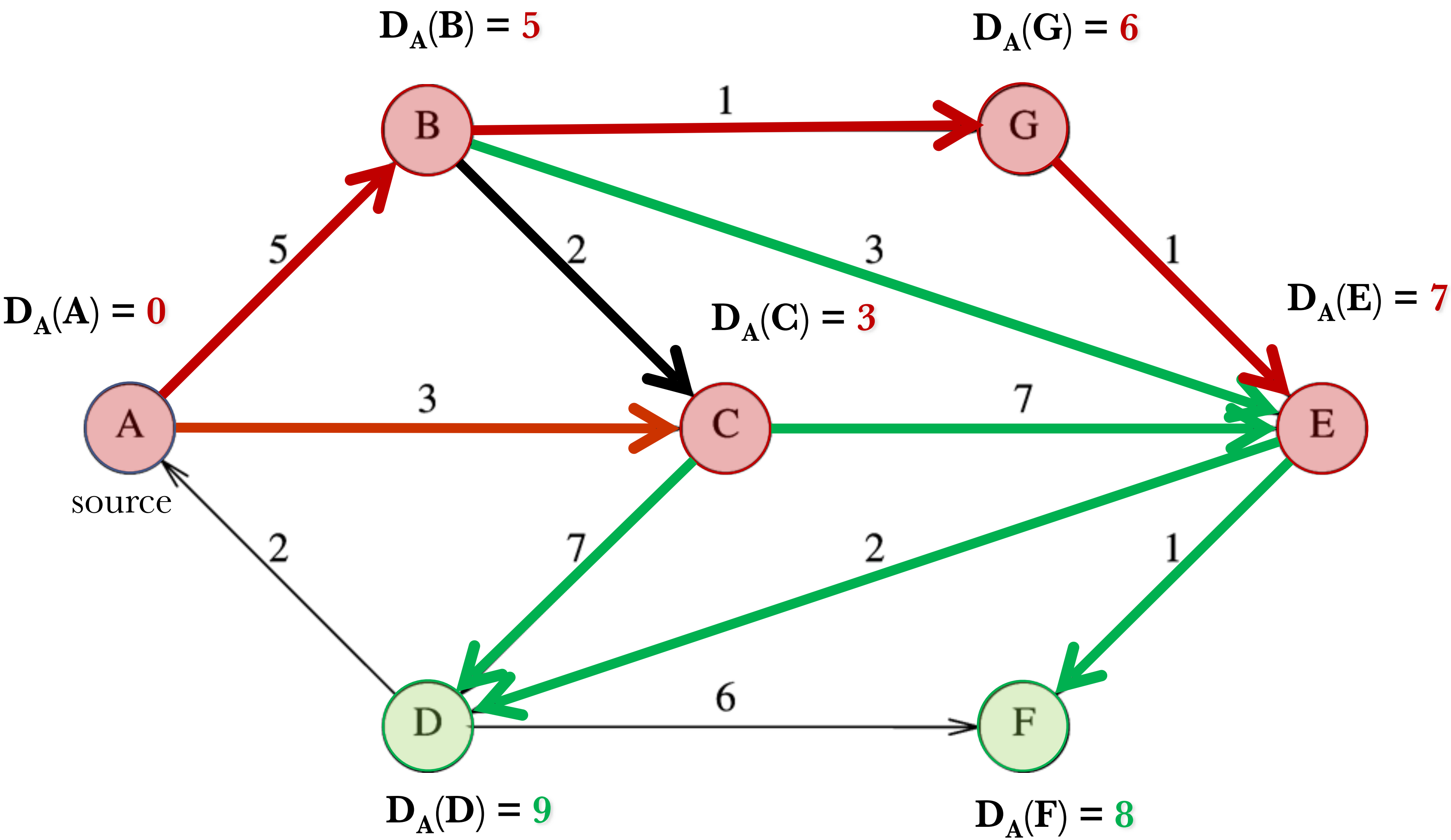
Weight $f(n) = D_{source}(Destination)$



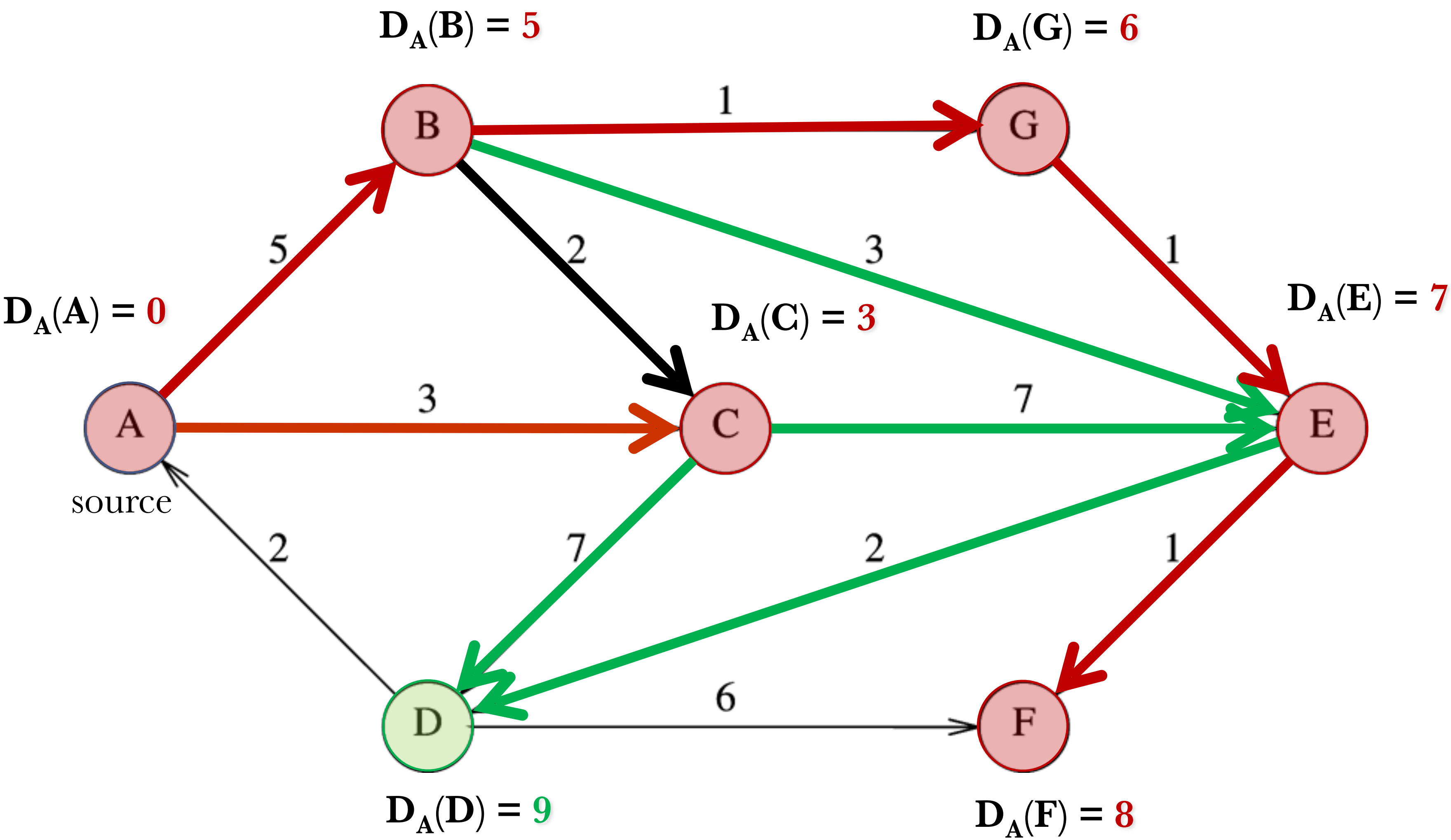
Weight $f(n) = D_{source}(Destination)$



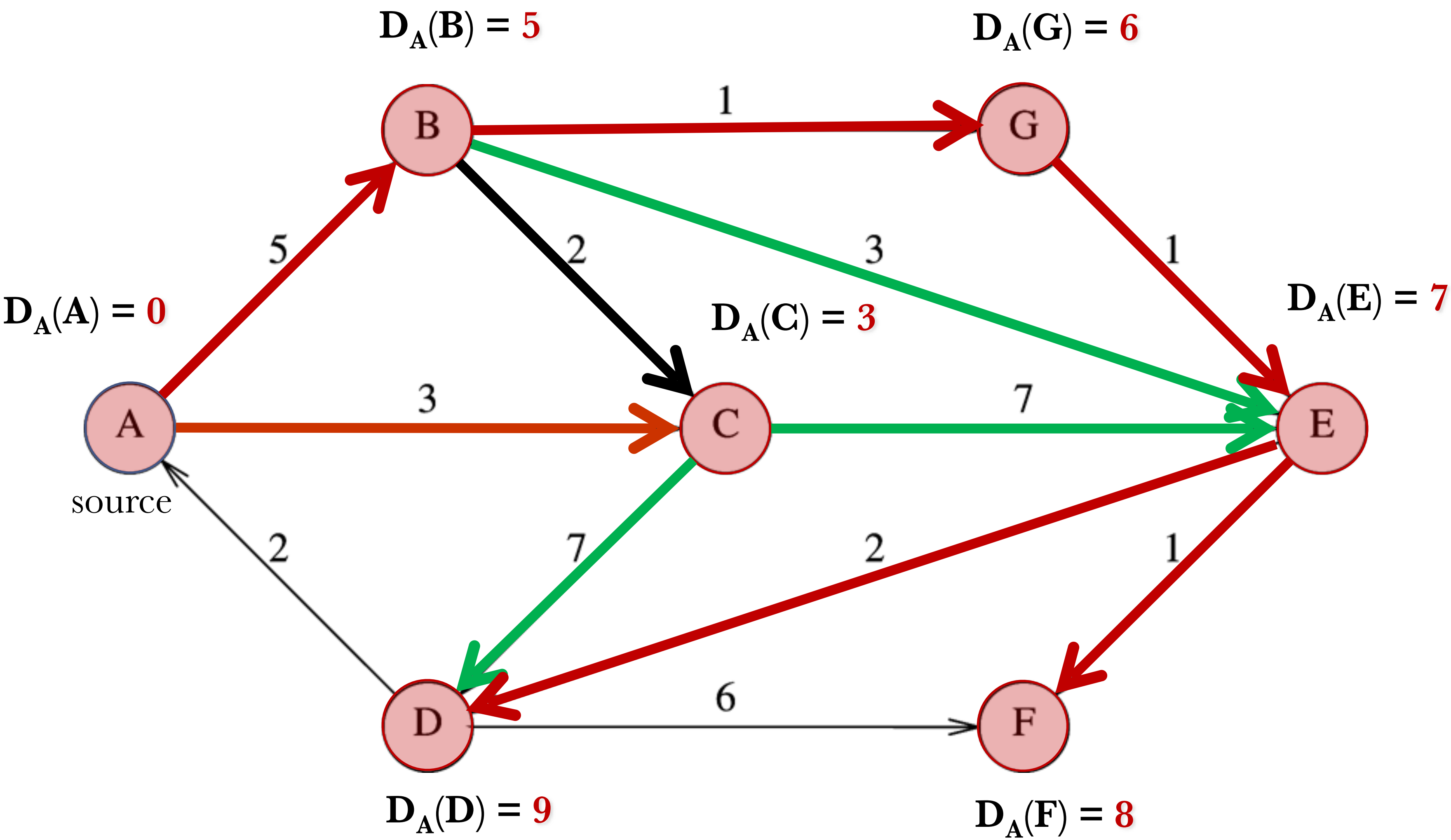
Weight f(n) = D_{source}(Destination)



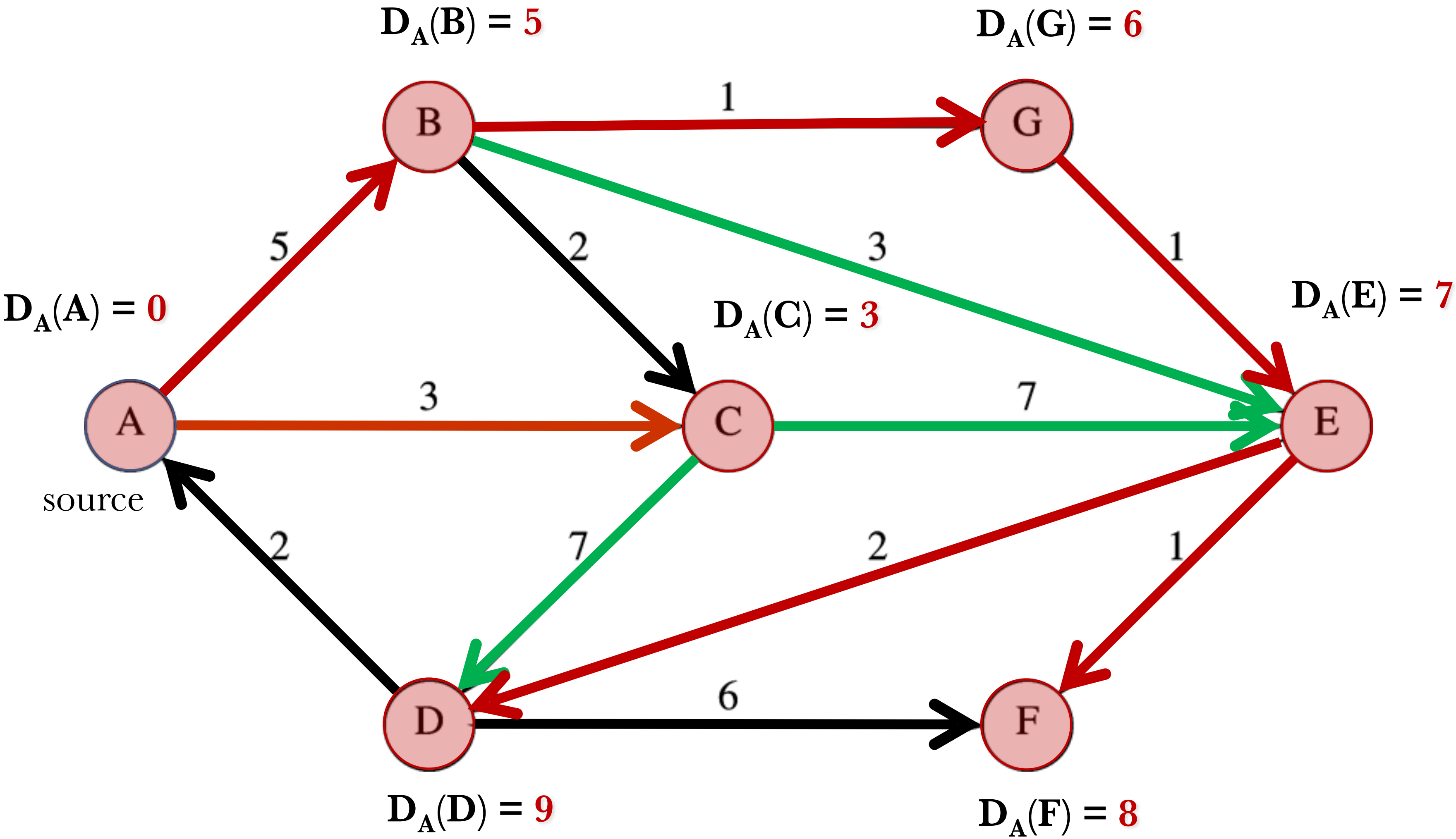
Weight $f(n) = D_{source}(Destination)$



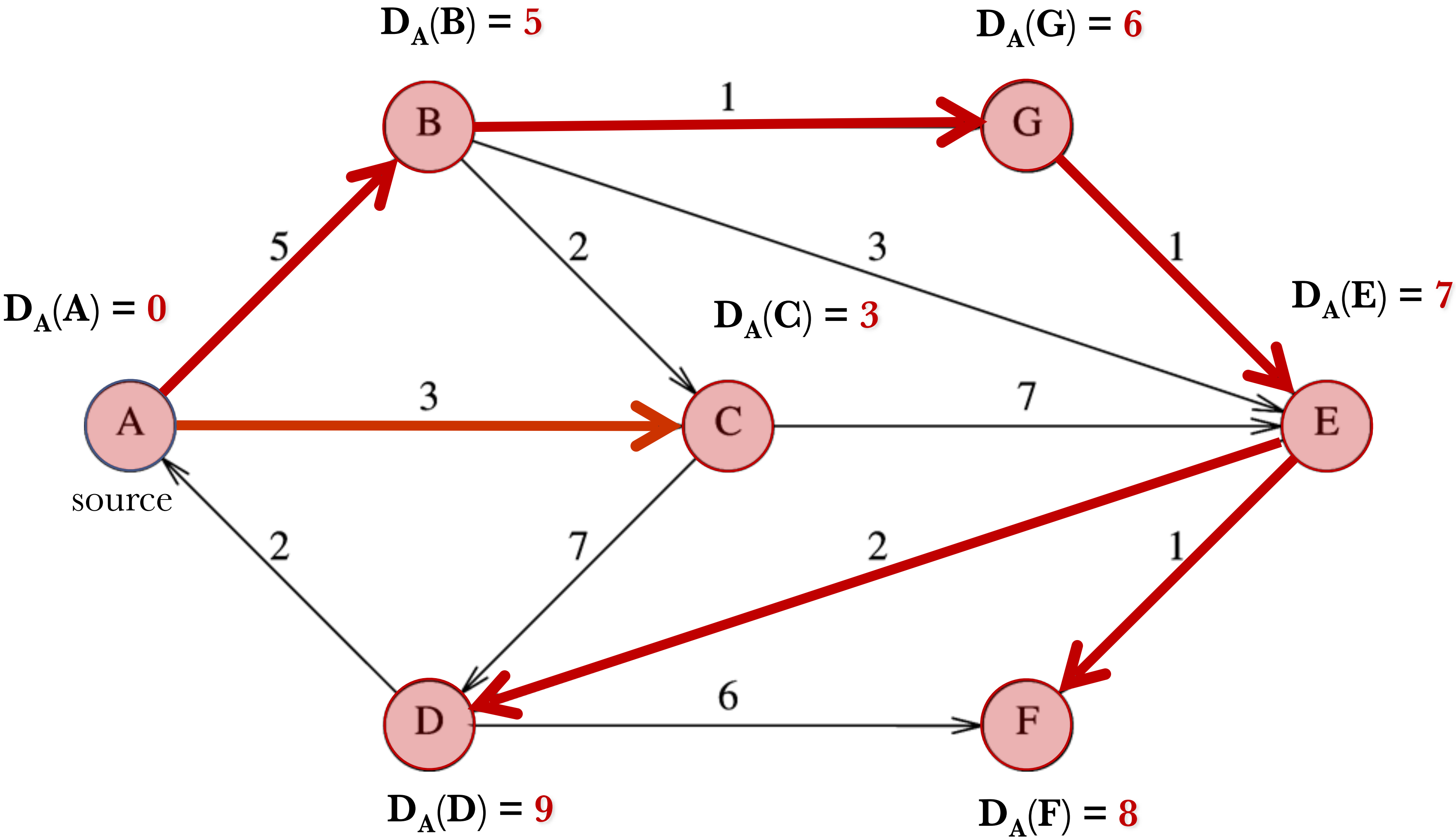
Weight f(n) = D_{source}(Destination)



Weight f(n) = D_{source}(Destination)

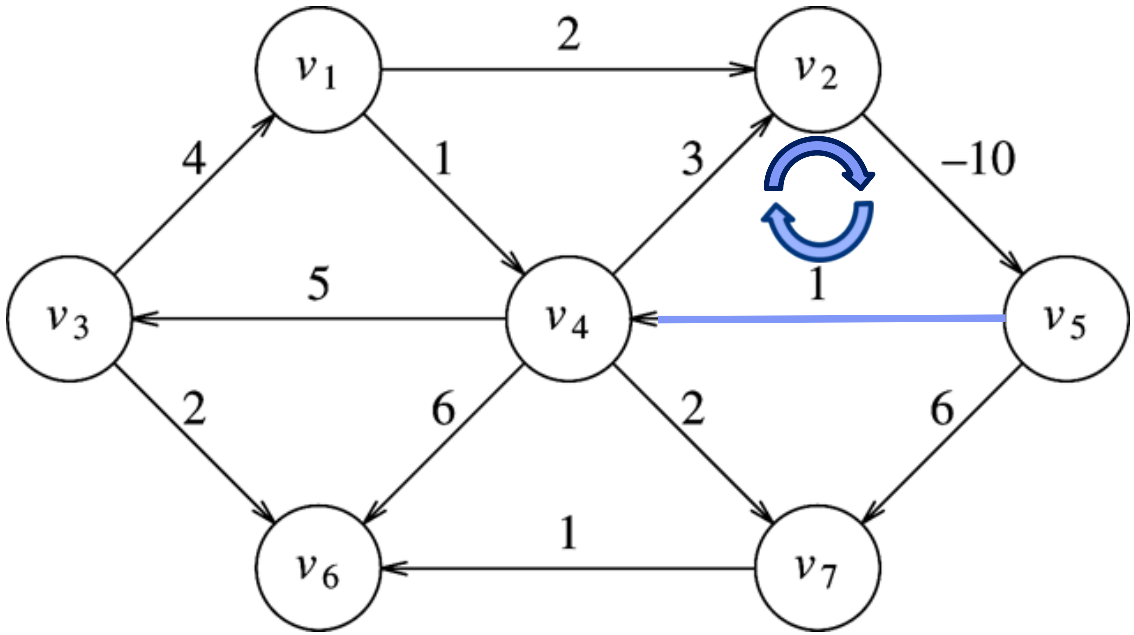


Weight $f(n) = D_{source}(Destination)$



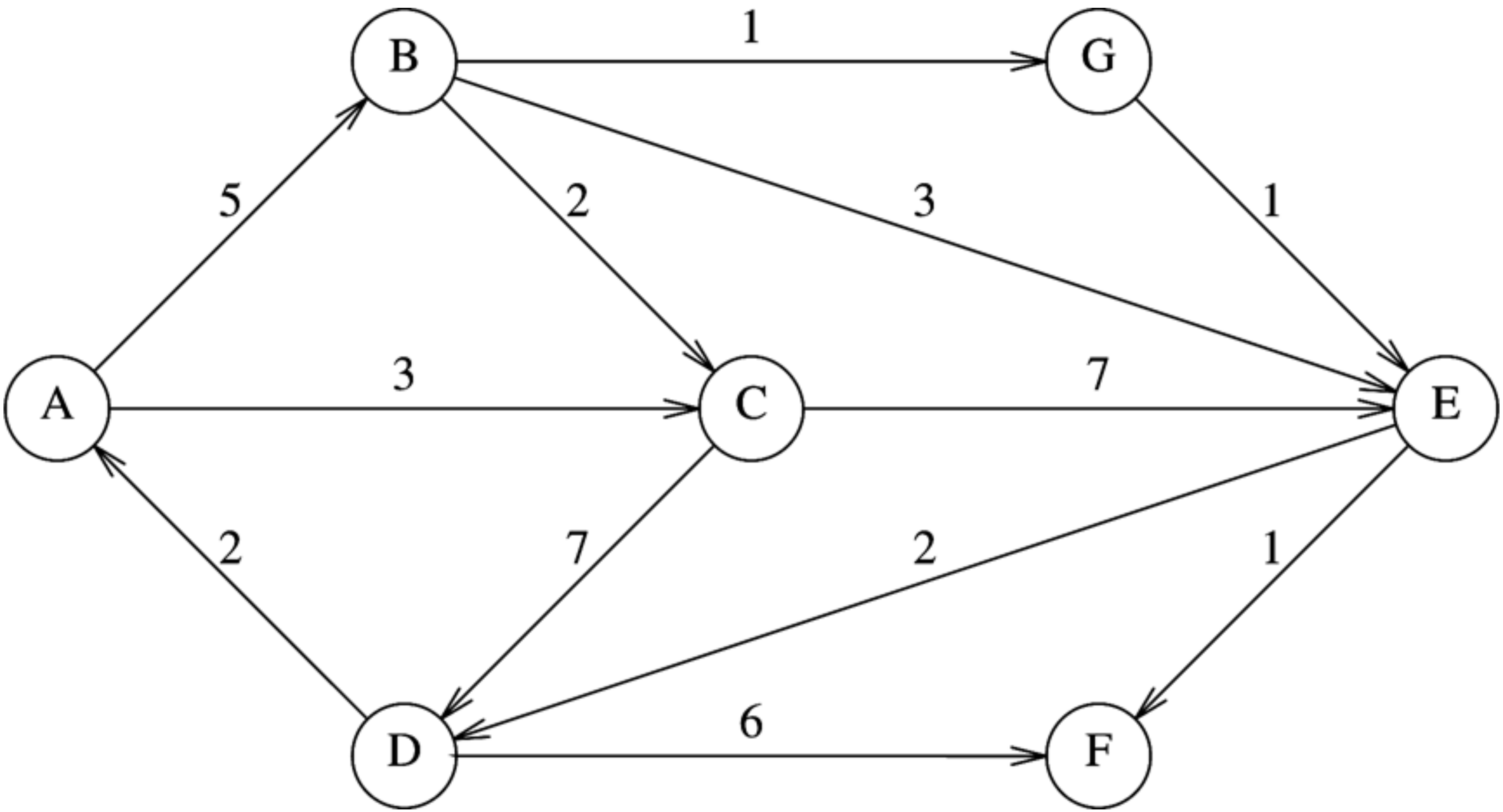
Weight $f(n) = D_{source}(Destination)$

Negative Cost Cycle



Dijkstra's does not work! cost = -∞

Which source vertex has no shortest path?



Dijkstra's Analysis

- Analysis reflects a complete execution of algorithm
- n = number of vertices, m = number of edges
- Initialize vertices is $O(n)$
- Finding the vertex with the minimum cost:
  - Use a list/array is $O(n^2)$, linear scan
  - Use a binary min-heap is $O(n\log n)$, deleteMin
- Update vertex's cost:
  - Use a list/array is $O(1)$
  - Use a binary min-heap is $O(m\log n)$, (percolateUp)
- Overall running time:
  - list/array is $O(n^2)$
  - binary min-heap is $O((n + m)\log n)$