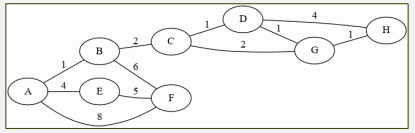
Shortest Weighted Path

BFS gives the shortest path in terms of the number of edges in the path

What if you have edge weights, and want the shortest path in terms of the sum of the edge weights along the path?

Shortest Weighted Path – Attempt

```
bfs(G,s)
  for each vertex u in V-{s} do
    state[u] = "undiscovered"
    prev[u] = null
    dist[u] = ∞
  state[s] = "discovered"
  prev[s] = null
  dist[s] = 0
  Q.enqueue(s)
  while Q is not empty do
    u = Q.dequeue()
    for each edge (u,v) in G.incidentEdges(u) do
      if state[v] = "undiscovered" then
        state[v] = "discovered"
        prev[v] = u
        dist[v] = dist[u] + w(u,v)  // +1 for bfs
        Q.enqueue(v)
    state[u] = "processed"
```



A	0			
B	∞	1		
C	∞	∞	3	
D	∞	∞		
E	∞	4		
F	∞	8	*	
G	∞	∞		
H	∞	∞		

- BFS sets dist[v] when a vertex is first discovered
- this label will be wrong if there is a longer path (more edges) with a lower cost

Shortest Weighted Path

Fix.

- allow dist[u] to be updated each time an edge (v,u) is encountered
- ensure that vertices along the shortest path s → u are processed before u
  - so that dist[u] won't need to be updated after u is processed

Implementation.

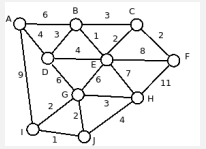
- observation: as long as all edge weights > 0, dist[v] < dist[u] for all vertices v on shortest path s → u
  - thus handling 'discovered' vertices in order of increasing dist[v] would satisfy the second part of the fix

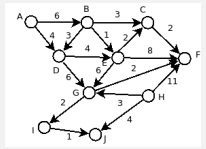
Dijkstra's Algorithm

finds the shortest path from *s* to every other vertex in the graph

to just find the shortest path from *s* to *t*, exit loop after *t* is removed from the PQ

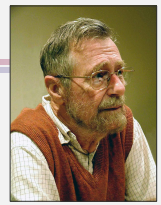
```
algorithm dijkstra(G,s):
  for all v in V do
    dist[v] ← ∞
    prev[v] = null

  dist[s] ← 0

  PQ ← makeQueue(V)

  while PQ is not empty do
    v ← PQ.removeMin()
    for each incident edge e=(v,u)
      if dist[u] > dist[v]+w(v,u) then
        dist[u] = dist[v]+w(v,u)
        PQ.decreaseKey(u)
        prev[u] = v
```





- can augment algorithm to store shortest path in addition to distance (add `prev` labels)

Edsger W Dijkstra



- Dutch computer scientist
- 1930 – 2002

- received the 1972 Turing Award for fundamental contributions to the development of structured programming languages
  - characteristics
    - three control structures: ordered sequence of statements, conditionals, loops
    - subroutines
    - blocks
  - in contrast to languages like BASIC which utilized GOTO statements

https://en.wikipedia.org/wiki/Edsger_W._Dijkstra