Shortest Paths Algorithms

Dijkstra’s algorithm for single-source shortest paths in graphs with non-negative weight

**DIJKSTRA(G,w,r)**

1. for each vertex \( u \in V[G] \)
2. do \( d[u] \leftarrow \infty \)
3. \( \pi[u] \leftarrow NIL \)
4. \( d[r] \leftarrow 0 \)
5. \( Q \leftarrow V[G] \)
6. while \( Q \neq \emptyset \)
7. do \( u \leftarrow EXTRACT-MIN(Q) \)
8. for each \( v \in \text{Adj}[u] \)
9. do if \( d[u] + w(u,v) < d[v] \)
10. then \( \pi[v] \leftarrow u \)
11. \( d[v] \leftarrow d[u] + w(u,v) \)
12. DECREASE-KEY(Q, v, \( d[v] \))

The implementation of Dijkstra’s algorithm uses a min-priority queue \( Q \), containing vertices \( v \) using \( d[v] \) as the key-value. The running time of Dijkstra’s algorithm depends on how we implement \( Q \). At the end of the algorithm’s execution, \( d[v] \) equals the total weight of a least-weight path from \( r \) to \( v \). This path can be obtained, in reverse order, by following \( \pi \) links from \( v \) to \( NIL \).

**FLOYD-WARSHALL(G,w,.)** (here \( w \) is a \( |V| \times |V| \) matrix)

1. \( d^0 \leftarrow w / \) (here \( d^k \) is a \( |V| \times |V| \) matrix, for \( k = 0,1,\ldots,|V| \))
2. for \( k \leftarrow 1 \) to \( |V| \)
3. for \( i \leftarrow 1 \) to \( |V| \)
4. for \( j \leftarrow 1 \) to \( |V| \)
5. \( d^k[i,j] \leftarrow \min (d^{k-1}[i,j], d^{k-1}[i,k] + d^{k-1}[k,j]) \)