

**Dijkstra Algorithm Handbook**

**Algorithmic Problem Solving**



**Prakash Hegade**

**1. Dijkstra’s Algorithm**

ALGORITHM Dijkstra(G, s)

// Dijkstra’s algorithm for single source shortest path

// Input: A weighted connected graph G(V, E) with non-negative weights and its vertex s

// Output: the length dv of a shortest path from s to v and its penultimate vertex pv for

every vertex v in V

Initialize(Q) // Initialize vertex priority queue to empty

for every vertex v in V do

dv 🡨 ∞

pv 🡨 null

Insert (Q, v, dv) // Initialize vertex priority in priority queue

ds 🡨 0

Decrease (Q, s, ds) // Update priority of s with ds

VT 🡨 Ø

for i 🡨 0 to |V| - 1 do

u\* 🡨 DeleteMin(Q)

VT = VT U {u\*}

for every vertex u in V – VT that is adjacent to u\* do

if du\* + w(u\*, u) < du

du 🡨 du\* + w(u\*, u)

pu 🡨 u\*

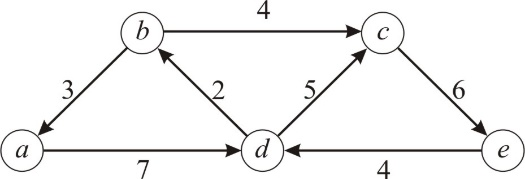
Decrease (Q, u, du)

**Intuition:**

Explain the algorithm in your own words by mapping it to tracing you did through notes. Also include the edge relaxation process.

Example: Find the shortest path form vertex c to all other vertices.

Vertices Mapping: {a:0, b:1, c:2, d:3, e:4}



cost [5][5] =

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** | **4** |
| **0** |  |  |  |  |  |
| **1** |  |  |  |  |  |
| **2** |  |  |  |  |  |
| **3** |  |  |  |  |  |
| **4** |  |  |  |  |  |

**Initialization:**

S = { 2 }

V – S = { 0, 1, 3, 4}

dist path

|  |  |
| --- | --- |
| 0 |  |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |

|  |  |
| --- | --- |
| 0 |  |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |

**Iteration 01:**

u = S =

dist[u] = V – S =

dist path

|  |  |
| --- | --- |
| 0 |  |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |

|  |  |
| --- | --- |
| 0 |  |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |

**Iteration 02:**

u = S =

dist[u] = V – S =

dist path

|  |  |
| --- | --- |
| 0 |  |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |

|  |  |
| --- | --- |
| 0 |  |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |

Working Space:

**Iteration 03:**

u = S =

dist[u] = V – S =

dist path

|  |  |
| --- | --- |
| 0 |  |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |

|  |  |
| --- | --- |
| 0 |  |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |

**Iteration 04:**

u = S =

dist[u] = V – S =

dist path

|  |  |
| --- | --- |
| 0 |  |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |

|  |  |
| --- | --- |
| 0 |  |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |

**Trace the path from vertex 2 to vertex 4:**