Proving Dijkstra's Algorithm correct.

We prove the iterative version of Dijkstra's Alg is correct by articulating our **Loop Invariant**, and proving (using Induction) that it is true at every iteration of the loop.

requirement for correctness

no obligations if pre-cond not met!

ShortestPaths(G, w, s)

// solves shortest paths from vertex s problem

// precondition: G is an undirected graph,
// w is a weight function on edges,
// $w: V(G) \times V(G) \to \mathbb{Z}^+$

// postcondition: d[v] = shortest s-v path weight
// prev[v] = previous-to-v vertex on shortest s-v path
// prev_w[v] = w(v, prev[v])

$H = \emptyset$
$X = \emptyset$
for each $v \in V(G)$, $v \neq s$
  $d[v] = \infty$
  insert v into H with key d[v]
$d[s] = 0$
insert s into H. with key d[s]
while H is not empty

// loop invariant: d[v] = min weight of any s-v path that only passes through vertices NOT in H. on way to v

loop

Extract v from H with smallest d[] value.
for each neighbour u of v:
  if $d[u] > d[v] + w(u,v)$
    $d[u] =$
    prev[u] =
    prev_w[u] =

// loop invariant: d[v] = min weight of any s-v path that only passes through vertices NOT in H. on way to v

Claim: the loop invariant is true at every iteration of the loop

First, let us show why the loop-invariant is exactly "n" what we need for correctness.

exit condition
+ loop invariant
post-condition

Claim: the loop invariant is true at every iteration of the loop

Proof: By induction on i, iteration number, $i = 0, 1, 2, \cdots n$