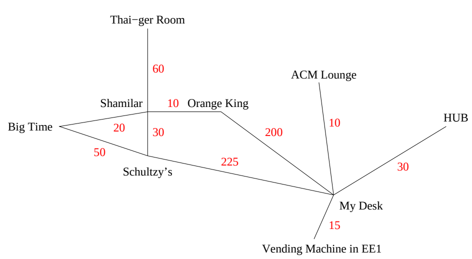
16—Shortest Paths and Dijkstra's Algorithm

May 17, 2002

Weighted Graphs

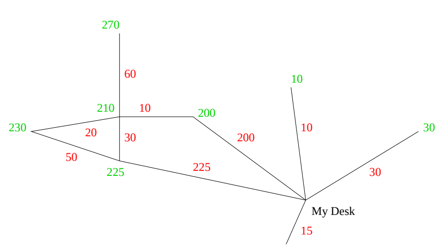


Quest for Food

(all distances in yards)

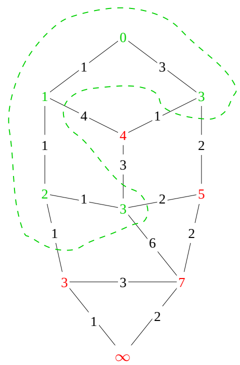
Serious About My Lunch

Single-Source Shortest Path



Compute shortest distance to all nodes from my desk

Dijkstra's Algorithm



Two kinds of verticies

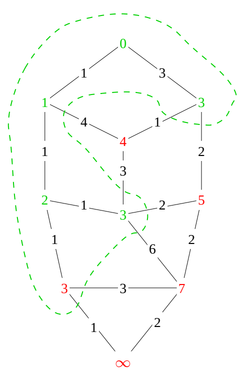
- Finished verticies

  Shortest distance computed

- Unknown verticies

  Have *tentative* distance

Dijkstra's Algorithm



At each step:

1. Pick *closest* unknown vertex
2. Add it to finished verticies
3. Update distances

Dijkstra's vs. BFS

At each step:

1. Pick closest unknown vertex
2. Add it to finished verticies
3. Update distances

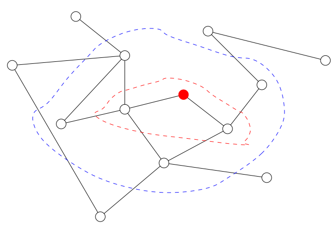
Dijkstra's Algorithm

At each step:

1. Pick vertex from queue
2. Add it to visited verticies
3. Update queue with neighbors

BFS

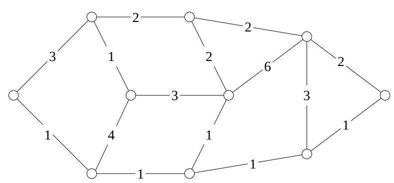
Grudge Match



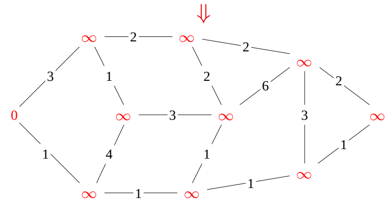
1. *Finished* verticies in the middle
2. Vectors from fringe added at each step

Dijkstra's and BFS

Yay! Example!

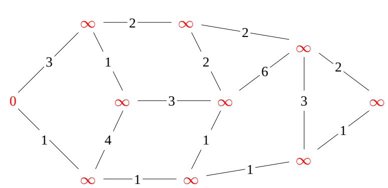


⇓
Initialize
⇓



- All verticies unknown
- Start vertex distance 0
- All other verticies at ∞

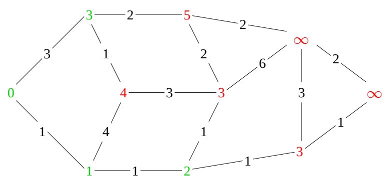
Examplo-rific



At each step:

1. Pick closest unknown vertex
2. Add it to finished verticies
3. Update distances

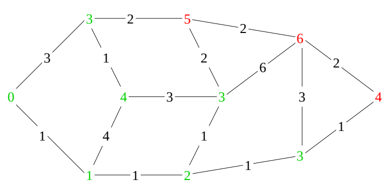
Examplarama



At each step:

1. Pick closest unknown vertex
2. Add it to finished verticies
3. Update distances

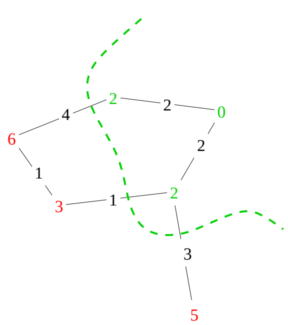
Examplophobia



At each step:

1. Pick closest unknown vertex
2. Add it to finished verticies
3. Update distances

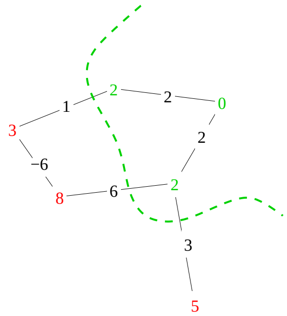
Would I Lie to You?



Why does this work?

Why are the distances of finished verticies correct?

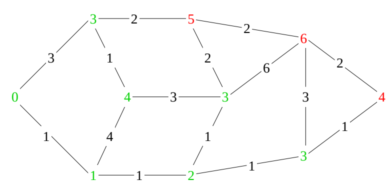
When Dijkstra's Wouldn't Work



Positive-only edge weights is essential

Key Lemma

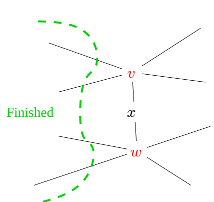
We finish with the closest verticies first



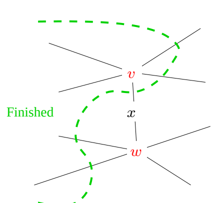
If w finished immediately after v, then
FinalDistance(w) $\geq$ FinalDistance(v)

Inductive Proof

If w finished immediately after v, then
FinalDistance(w) $\geq$ FinalDistance(v)



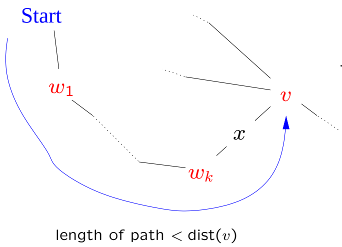
Tentative distance of w no bigger than v's



Updates from v may *shrink* distance of w, but can't get smaller than v's distance

The Big Kahuna

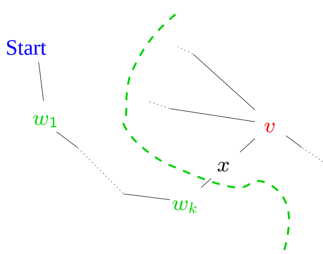
Suppose v's final distance is *not* correct



Then exists a shorter path to v

- dist(w_k) + x < dist(v)
- Final distances on path are correct
- v finalized *after* all other nodes on the path

Gotcha!

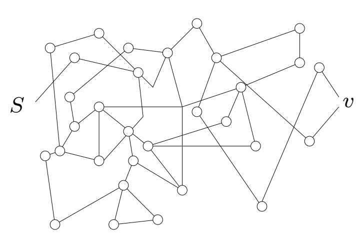


When w_k finalized, dist(v) ← dist(w) + x

- Final dist(v) *greater* than this
- dist(v) only *decreases* during the algorithm
- Contradiction!

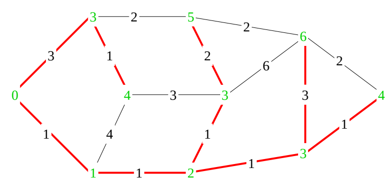
A Number is Not a Path

Sure, we know how far away v is. . .



. . . but how do we *get* there?

Shortest Path Tree

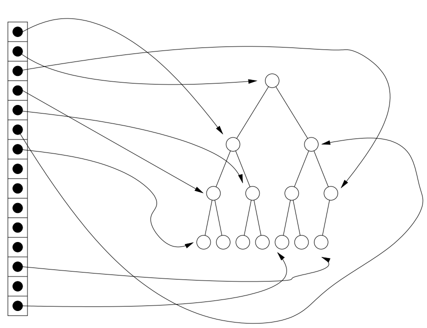


Analogue of BFS tree

In All Its Glory

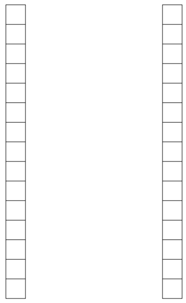
```
Dijkstra(Graph G, Vertex s)
{
   PQ pq;
   for each (v in G) pq.Insert(v, INFINITY);
   pq.Change(s, 0);
   while (!pq.Empty()) {
      v = pq.DeleteMin();
      for each (w in v.Neighbors())
        if (dist(w) > dist(v) + w(v,w))
          pq.Change(w, dist(v)+w(v,w));
   }
}
```


Implementing the PQ



The List of Vertices The PQ

PQ as a Heap



vtx_loc heap

```
PQ::SwapUp(i)
{
  int p = (i-1)/2;
  while (p >= 0
         && heap[p].key
             > heap[i].key) {

    swap(heap[p],heap[i]);



    i = p;
    p = (i-1)/2;
  }
}
```