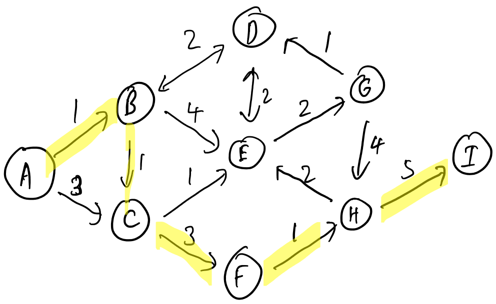
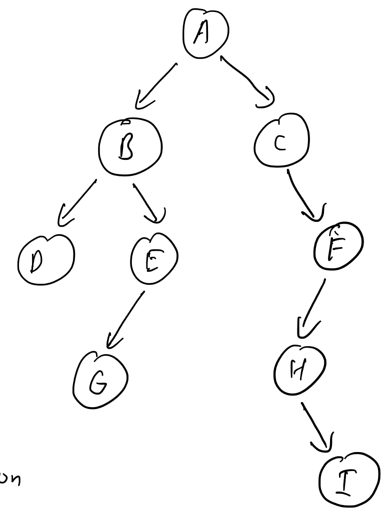
Lec08-Dijkstra (1959) KT 4.4

Wednesday, September 11, 2024 3:22 AM

Shortest path in weighted, directed graph from starting node

(HW problem is about counting shortest paths in unweighted graph)





BFS works for unweighted graph

Applications:

- Online maps & GPS navigation
- Routing systems (packets on networks)
- Drug-Pathway interactions
- Epidemiology (path of disease through social network)

Dijkstra: (1959)

Use tree-growing paradigm (start w/ node; keep adding frontier edge)

Recall: Prim's alg: nextedge = frontier edge of min weight

Shortest path

- Let s be the starting node
- Maintain an array $d[u]$ = length of shortest currently known $s \to u$ path. ($d[s]=0$)
- nextedge = frontier edge (u,v) that minimizes $\underbrace{d[u]}_{\text{modification from Prim's}} + \text{length}(u,v)$.

Pseudocode:

for $u \in V$, $\underbrace{d[u]=\infty}_{\text{tentative distance}}$, $\underbrace{p[u]=\emptyset}_{\text{tentative parent}}$, $\underbrace{F=V}_{\text{frontier}}$

F = makeHeap(V, d)

for $u \in V$, $d[u] = \infty$, $p[u] = \emptyset$, $F = V$

F = makeHeap(V, d)

$d[s] = 0$

while $F \neq \emptyset$:

  $u \leftarrow$ vertex in F with min $d[u]$ } — deleteMin() in heap — always explore next nearest node

  remove u from F

  for each neighbor v of u in F:

    if $d[u] + \text{length}(u,v) < d[v]$: } reduce tentative distances + choose new parent

      $p[v] = u$

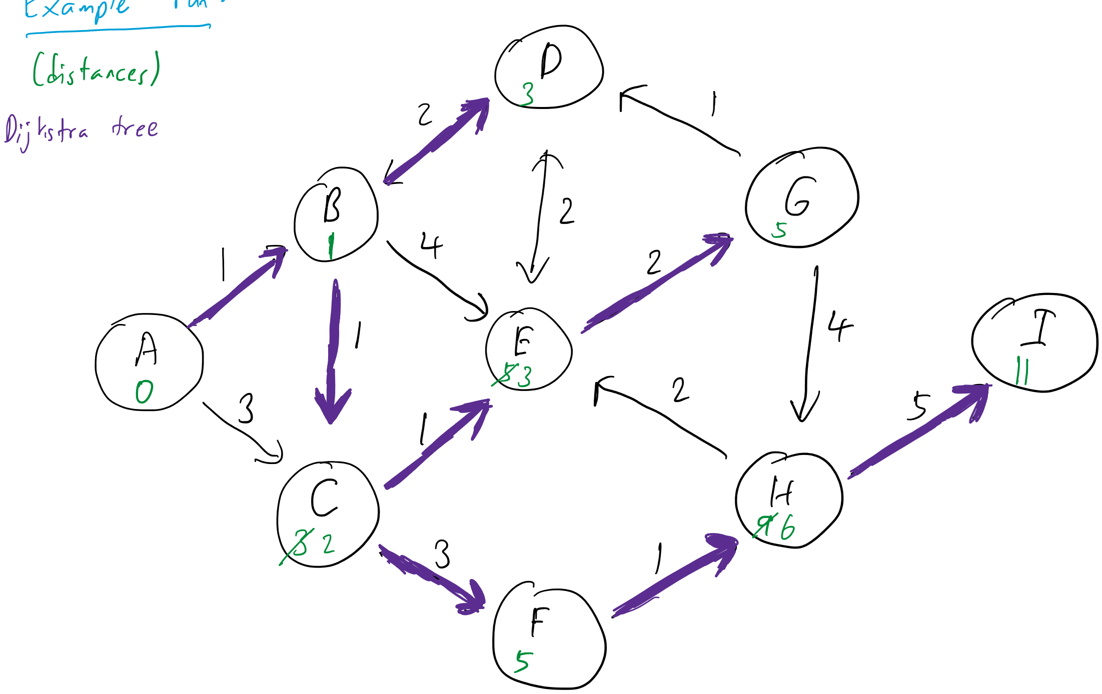
      $d[v] = d[u] + \text{length}(u,v)$ — reducekey of heap

return $d[u]$, $p[u]$

Example run:

(distances)

Dijkstra tree



Correctness: Let T be the set of nodes explored at some point during the alg.

$\forall u \in T$, the path found by Dijkstra's is the shortest

proof. By induction on the size of T.

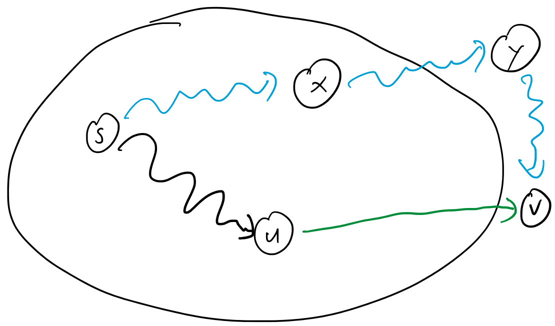
Base case: $|T| = 1$, so $|T| = \{s\}$, $d[s] = 0$, so correct.

Induction hypo: Assume theorem is true when $|T| \leq k$.

Let v be the $(k+1)$st node added using edge $\underline{(u,v)}$.

Let P_v be the path chosen by Dijkstra.

Let P' be any other $s \to v$ path. (blue)



length $(s \to u \to v) \le$ length (blue) by design of alg, since otherwise would have added y before adding v.

So, induction works.

$\boxed{\diagup}$

Thm $\exists$ optimal set of shortest paths from s and their union is a tree

proof. Dijkstra works.

Runtime: Same as Prim's MST

- every edge is processed once
  - └ Either:
    - (1) do nothing: $O(1)$
    - (2) reduce key in heap ; $O(\log|V|)$
- ~~total~~ time : $O(|E| \log |V|)$