

<http://www.cs.unh.edu/~ruml/cs758>

Graph Traversal

- Graphs
 - Breadth-first
 - The Algorithm
 - Full Algorithm
 - Break
 - Factoids
 - Proof
 - Depth-first Search
 - EOLOs
 - Edges
- Graph Traversal

Graphs

directed, arc/edge, weighted, labeled, drawings

representation

relations \rightarrow edges

cycle, DAG, tree, planar

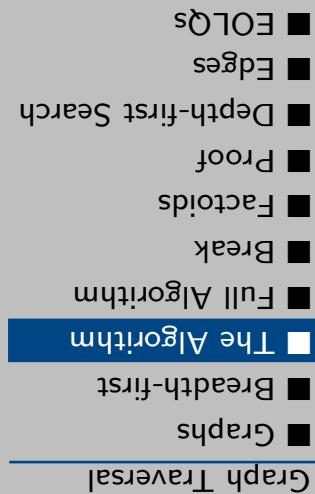
- Graphs
- Breadth-first
- The Algorithm
- Full Algorithm
- Break
- Factoids
- Proof
- Depth-first Search
- Edges
- EOLQs

Breadth-first Search

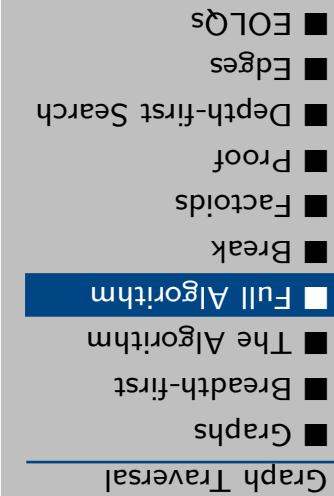
traversal: graph \rightarrow forest

- Graphs
- Graph Traversal
- Breadth-first
- The Algorithm
- Full Algorithm
- Break
- Factoids
- Proof
- Depth-first Search
- Edges
- EOLQs

- Graph Traversal
- 1. foreach vertex, label it undiscovered and $u.d \rightarrow \infty$
 - 2. start's label \rightarrow discovered, $d \rightarrow 0$, $\pi \rightarrow \text{nil}$
 - 3. $Q \rightarrow \{\text{start}\}$
 - 4. while Q not empty
 - 5. $u \rightarrow \text{dequeue}(Q)$
 - 6. foreach neighbor v of u
 - 7. if v is undiscovered
 - 8. label v discovered, $v.d \rightarrow u.d + 1$, $v.\pi \rightarrow u$
 - 9. enqueue v in Q
 - 10. label u finished
- Which vertices does Q hold (at line 4)?
- Do we really need all the labels?
- What's the time complexity?



- What's the time complexity?
1. foreach vertex, label it undiscovered and $u.d \rightarrow \infty$
 2. foreach vertex s
 3. if s is undiscovered
 4. $s.label \rightarrow \text{discovered}$, $s.d \rightarrow 0$, $s.\pi \rightarrow \text{nil}$
 5. $\emptyset \rightarrow \{s\}$
 6. while \emptyset not empty
 7. $u \rightarrow \text{dequeue}(\emptyset)$
 8. foreach neighbor v of u
 9. if v is undiscovered
 10. label v discovered, $v.d \rightarrow u.d + 1$, $v.\pi \rightarrow u$
 11. enqueue v in \emptyset
 12. label u finished

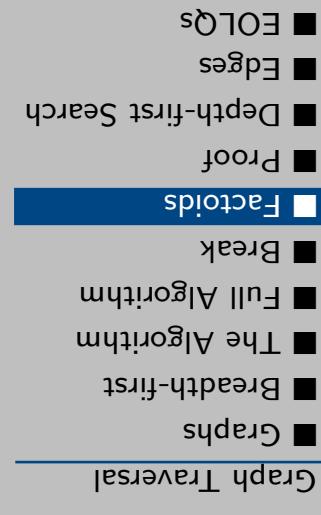


- schedule: no class next Tue
- midterm
- ass't 7
- Breadth-first
- The Algorithm
- Full Algorithm
- Break

- Graph Traversal
- Graphs
- Breadth-first
- The Algorithm
- Full Algorithm
- Break
- Factoids
- Proof
- Depth-first Search
- Edges
- EOLQs

Break

1. Distances we assign always stay the same or go down.
2. $u.d \geq f(s, u)$
- Proof: Show $u.d \geq f(s, u) \forall u$ via induction over iterations:
Holds at start.
 $u.d$ is updated to $u.d + 1 \geq f(s, u) + 1 \geq f(s, u)$.
3. d values in queue are nondecreasing and last in queue exceeds first by at most 1.
Proof: By induction. True when queue is s .
Preserved by dequeue.
Enqueue: $\text{new}.d = \text{removed}.d + 1 \leq \text{first}.d + 1$ and
 $\text{last}.d \leq \text{removed}.d + 1 = \text{new}.d$.
4. At termination, $u.d = f(s, u)$ = shortest path length



Claim: at termination, $u.d = \delta(s, u)$ = shortest path length

Consider u with minimum incorrect distance, and u that is before it on a shortest path. $u.d > \delta(u) + 1 = u.d + 1$. When u is dequeued:

- if u is undiscovered, it would then be correct, contradiction.
- if u is already finished, then $u.d \leq u.d$, contradiction.
- if u is discovered, let w be predecessor. $u.d = w.d + 1$ and $w.d \leq u.d$ so $u.d \leq u.d + 1$, contradiction.

Depth-first Search

DFS
1. forall vertices, label \rightarrow undiscovered

2. DFS-visit(*start*)

DFS-visit(*u*)

3. label *u* discovered

4. foreach neighbor *v* of *u*
5. if *v* is undiscovered

6. $v.\pi \rightarrow u$

7. DFS-visit(*v*)

8. label *u* finished

Vs breath-first?

Discovery and finish times are parenthesized

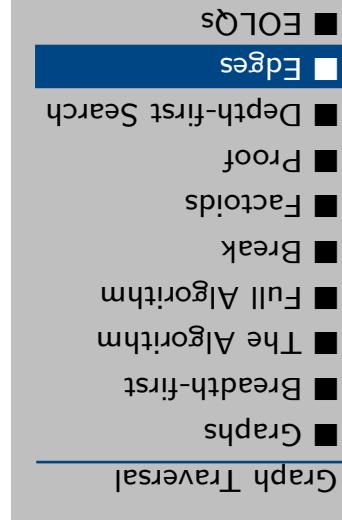
What's the time complexity?

- Graphs
- Breadth-first
- The Algorithm
- Full Algorithm
- Break
- Factoids
- Proof
- DFS-visit Search
- Degrees
- EOLQs

Edges

tree: in depth-first tree
back: connects to ancestor in tree
forward: non-tree edge connecting to descendant in tree
cross: others: non-ancestors/non-descendants or different DFS
tree

when edge is explored, label of arc dest gives type



EOLQs

For example:

■ What's still confusing?

- What question didn't you get to ask today?
- What would you like to hear more about?

Please write down your most pressing question about algorithms
and put it in the box on your way out.

Thanks!

Graph Traversal	EOLQs
■ Breadth-first	■ EOLQs
■ Graphs	■ Edges
■ The Algorithm	■ Depth-first Search
■ Full Algorithm	■ Proof
■ Break	■ Factoids
■ Factorids	■ Depth-first
■ Fac	■ EOLQs