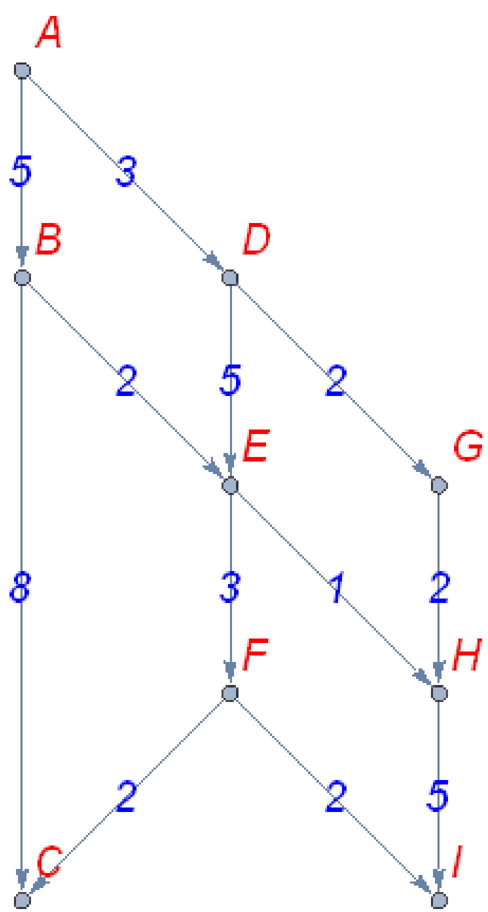
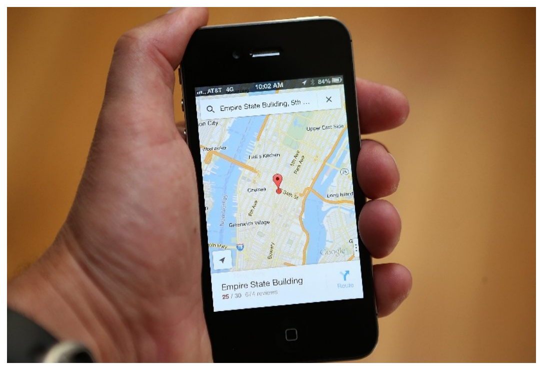
Greedy: Graph Exploration

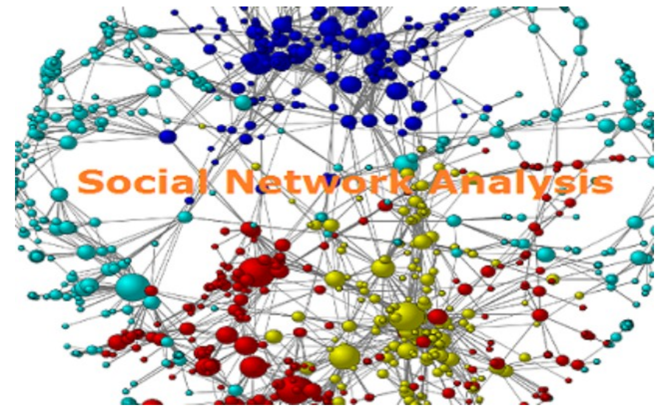
SSSP

Single-source Shortest Path Problem (SSSP): Given a source vertex v in a directed graph, find the shortest path from v to every other vertex in the graph (assuming it exists).

Applications: driving apps, video games, social networks, network broadcasting

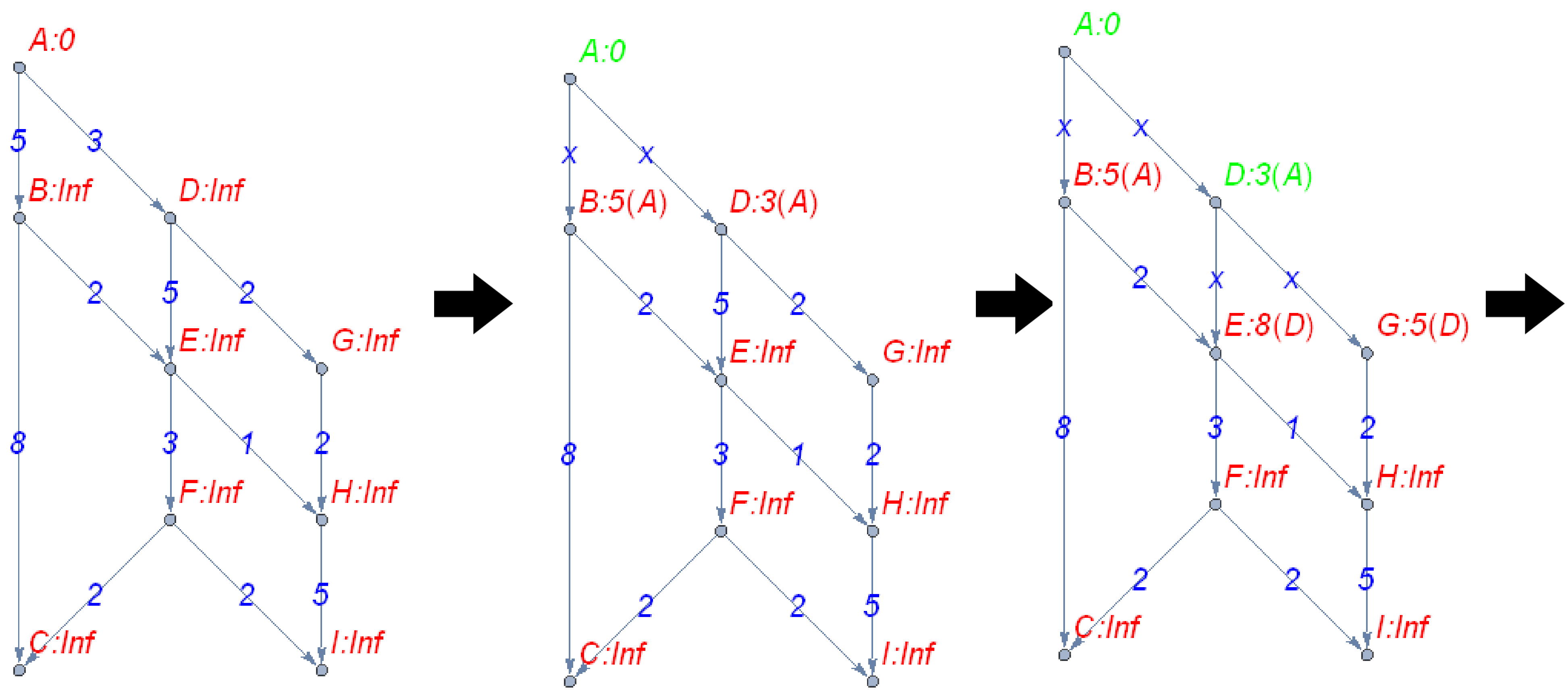




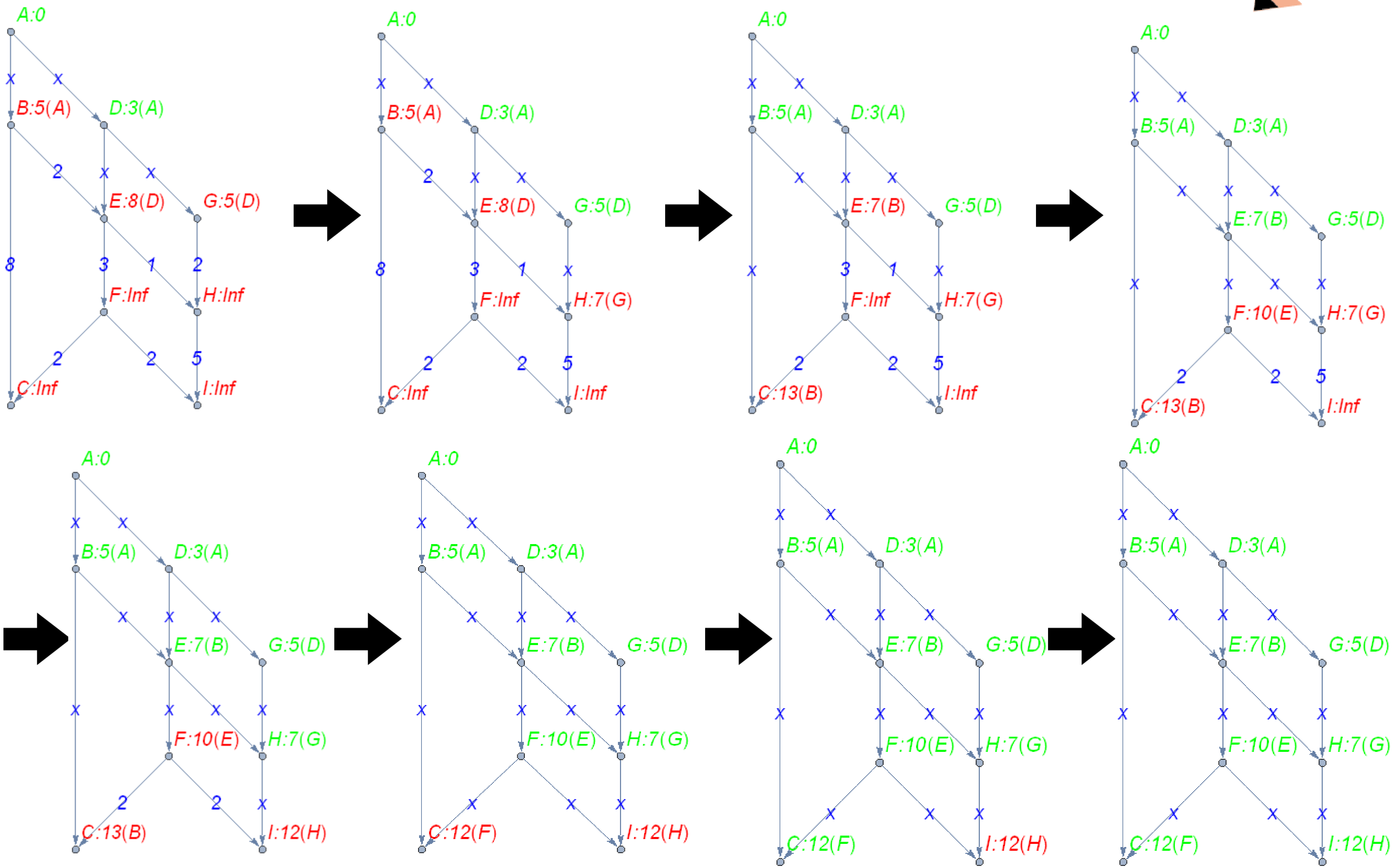


Dijkstra’s Algorithm

<u>Dijkstra’s Algorithm</u>: Initialize the scores of all vertices to ∞ and the score of the source to 0. Mark all vertices unfinished. While there are still unfinished vertices, find the vertex with smallest unfinished score, mark it finished, and relax all of its outgoing edges.



Dijkstra's Algorithm

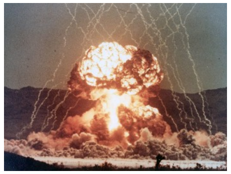


Dijkstra's Algorithm

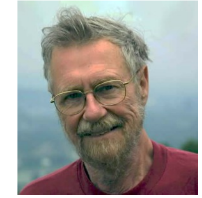
Claim: Every time that a vertex v is marked finished, $score(v) = d(source, v)$.
Proof: Is this true when the algorithm begins? Yes.
Assume that up to a certain point in Dijkstra's Algorithm, all vertices that have been marked finished satisfy $score(v) = d(source, v)$ and that we are about to mark a vertex a finished and $score(a) \neq d(source, a)$.
Is it possible for $score(a) < d(source, a)$? No. If that were the case, we would have discovered a path from $source$ to a shorter than $d(source, a)$.
$\Rightarrow score(a) > d(source, a)$
Is it possible for all vertices along a shortest path from $source$ to a to be marked finished? No. If that were the case, then the score on a would be $d(source, a)$.
Let x be the first vertex along a shortest path from $source$ to a that is *not* marked finished.

$source$ (finished) $\rightsquigarrow x$ (not finished) $\rightsquigarrow a$ (about to be marked finished)

Note that because all vertices along the shortest path from $source$ to x are finished, the score on x must be $d(source, x)$.
$score(x) = d(source, x) < d(source, a) < score(a)$
Why aren't we marking x (instead of a)?



Edsger Dijkstra



When Dijkstra married (1957) in the Netherlands, the marriage rites required him to state his profession. When he stated that he was a computer programmer, the authorities objected that there was no such profession. His marriage certificate instead identifies him as a theoretical physicist.

Dijkstra (one of the authors of Algol-60) introduced recursion into the language via the “stack.” In the Oxford English Dictionary, the terms “vector” and “stack” are attributed to Dijkstra in the computing context.

Dijkstra (1968) argued that the GOTO statement should be eliminated. Neither Java nor Python supports a GOTO statement now.

Dijkstra won the 1972 Turing Award.

The question of whether computers can think is like the question of whether submarines can swim.

Brainpower is by far our scarcest resource.

Program testing can at best show the presence of errors but never their absence.

Being abstract is something profoundly different from being vague.

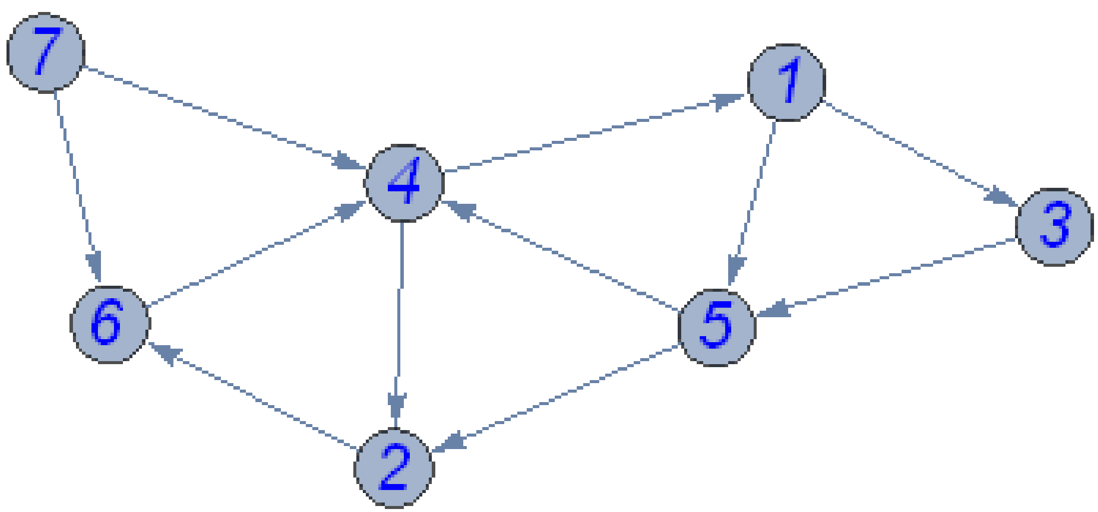
If 10 years from now, when you are doing something quick and dirty, you suddenly visualize that I am looking over your shoulders and say to yourself, Dijkstra would not have liked this, well that would be enough immortality for me.

Depth First Search

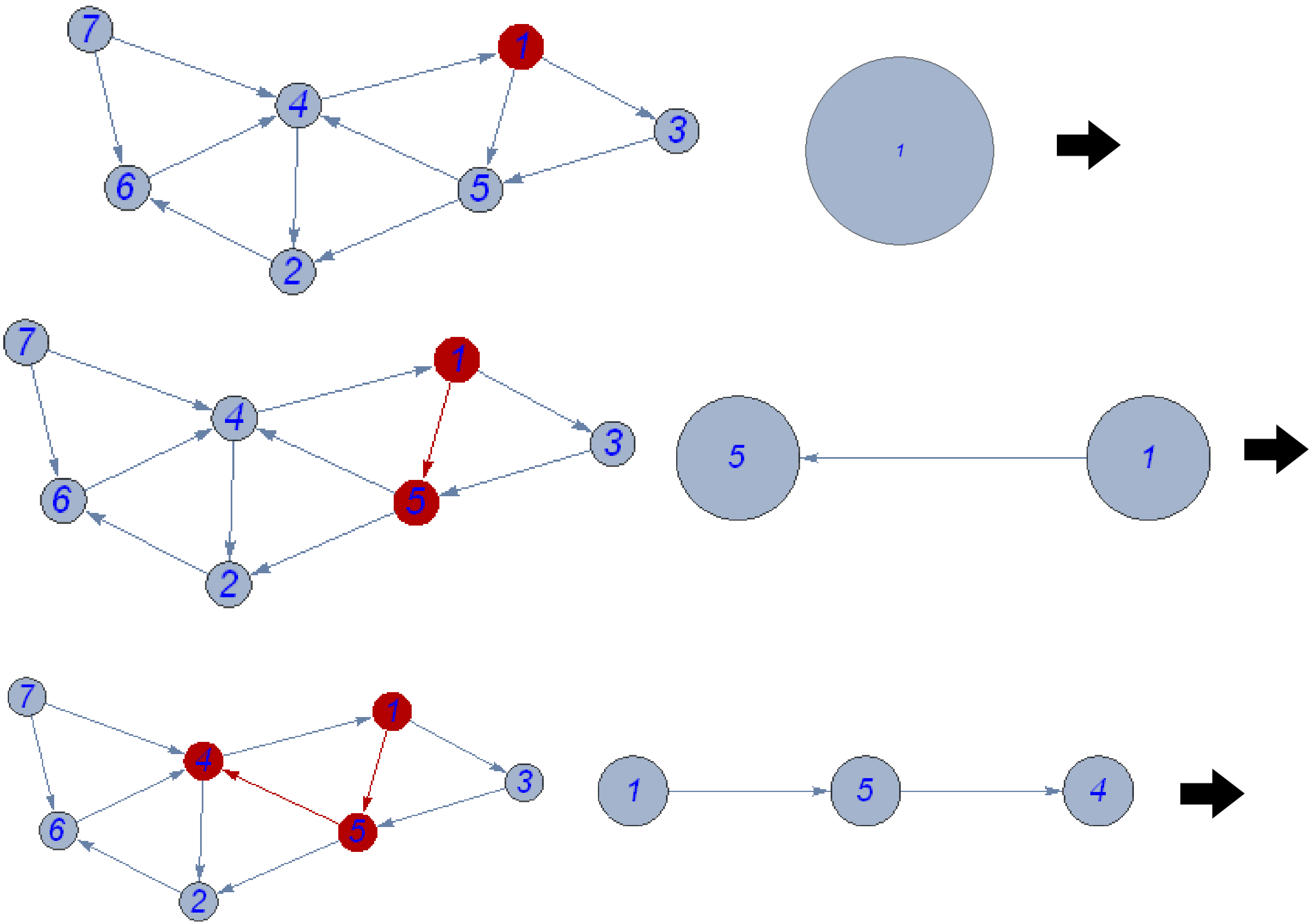
Problem: How can we explore a network efficiently?

The basic idea is that you are only allowed to "see" locally, and your goal is to map the global environment.

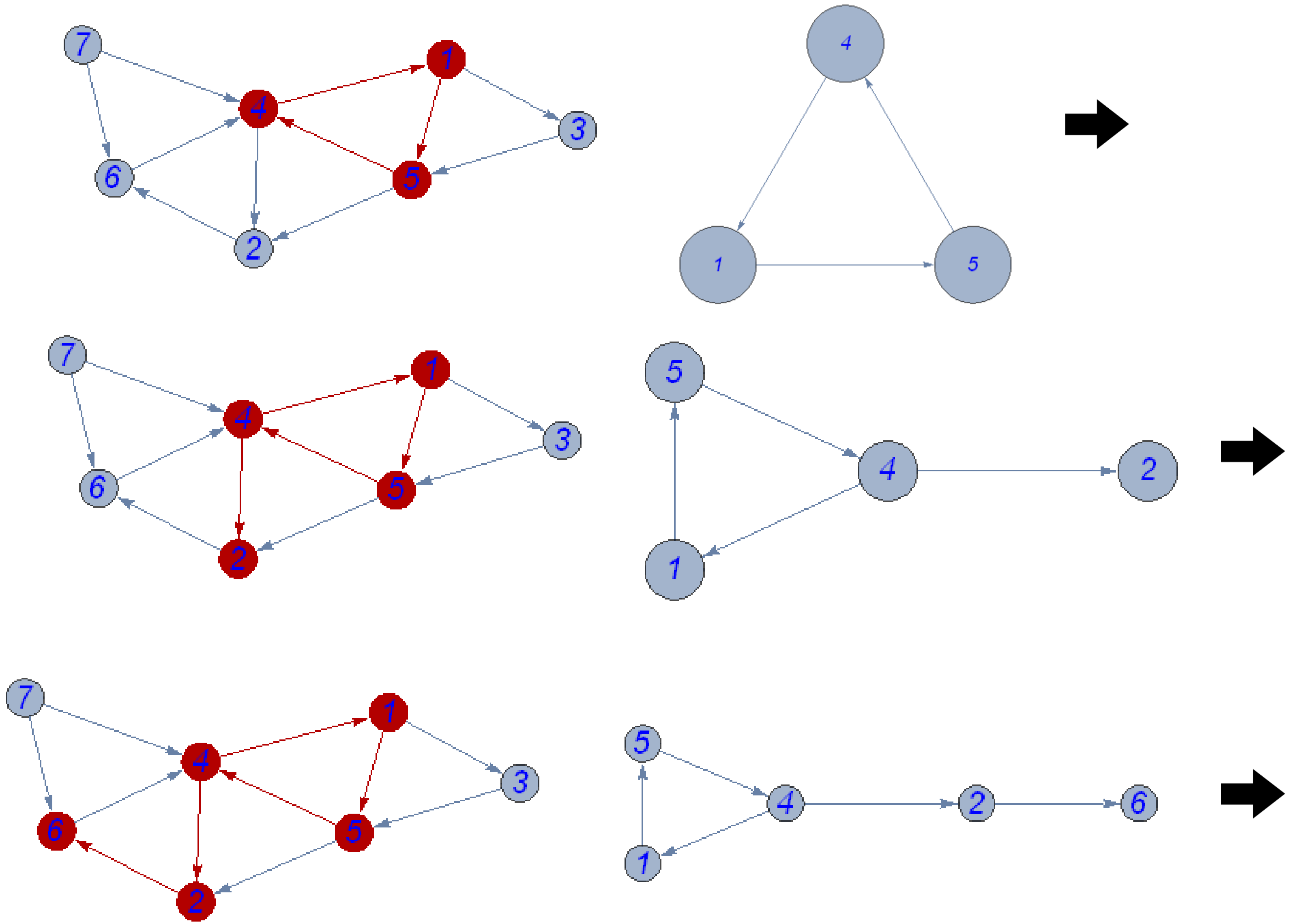
Depth First Search (DFS): While there are still unexplored vertices in the graph, teleport to any unexplored vertex and mark it explored. While there are unexplored outgoing edges from the current vertex, follow/map the edge. If you encounter a vertex that you have already explored, back up. If there are no unexplored outgoing edges, mark the current vertex finished and back up/teleport.



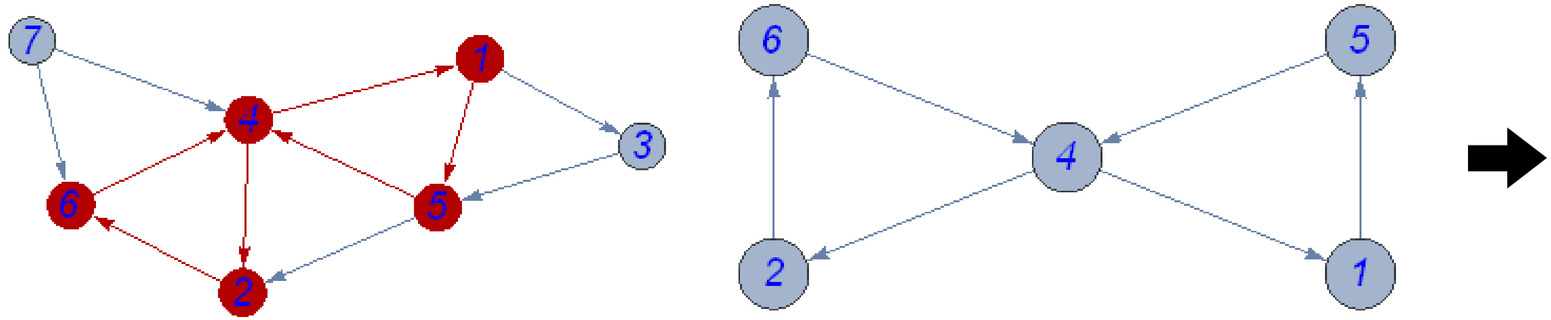
DFS: Example



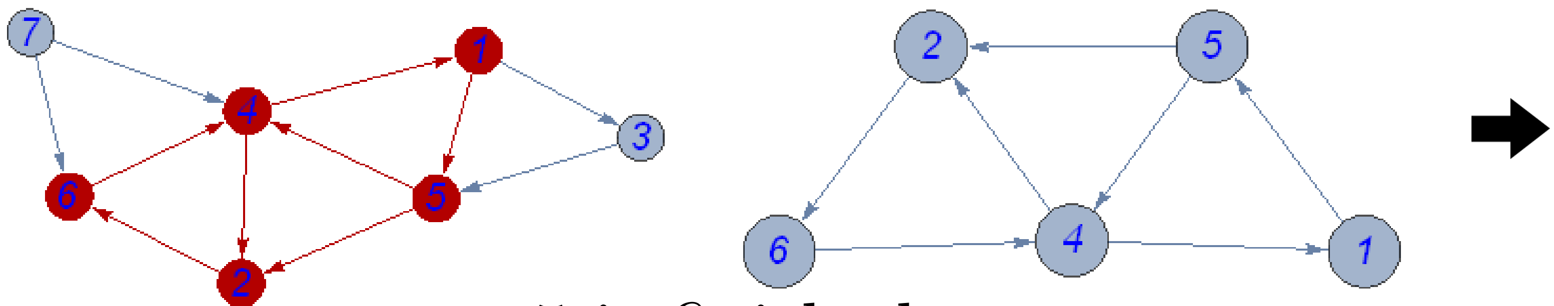
DFS: Example



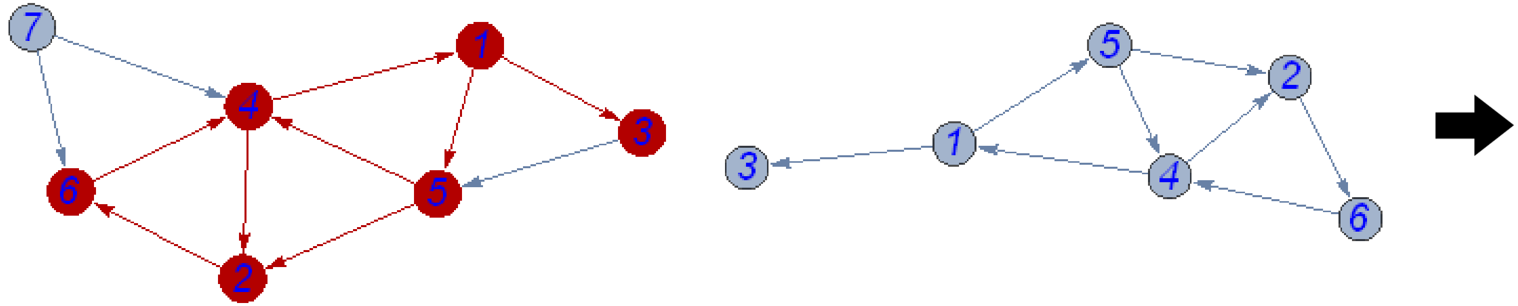
DFS: Example



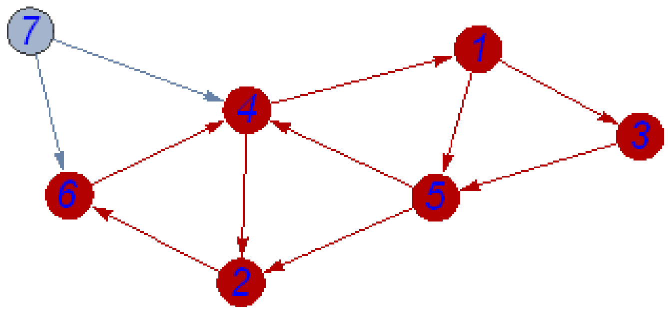
6 is finished. 2 is finished. 4 is finished.

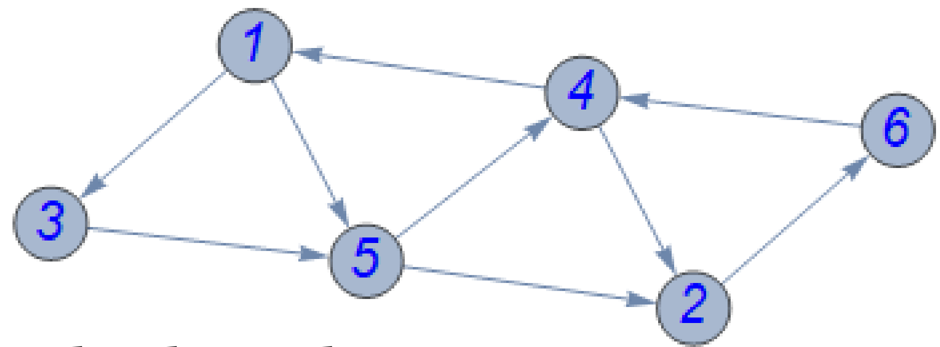


5 is finished.

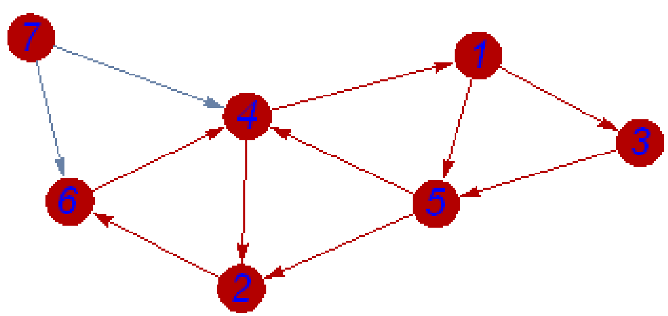


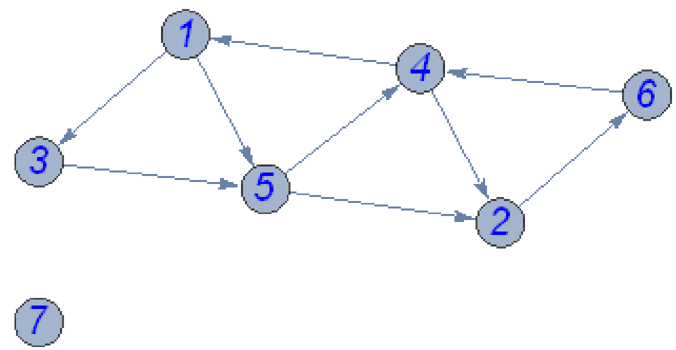
DFS: Example

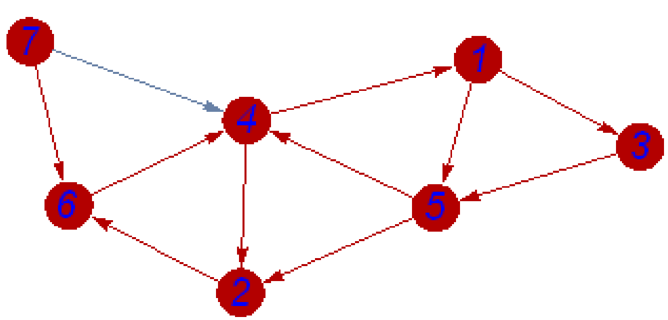


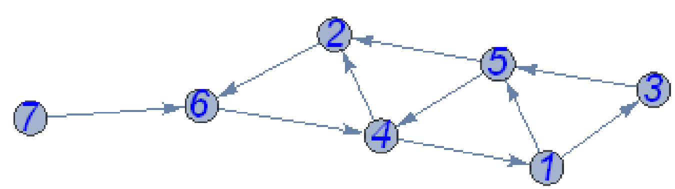


3 is finished. 1 is finished. Teleportation to 7.

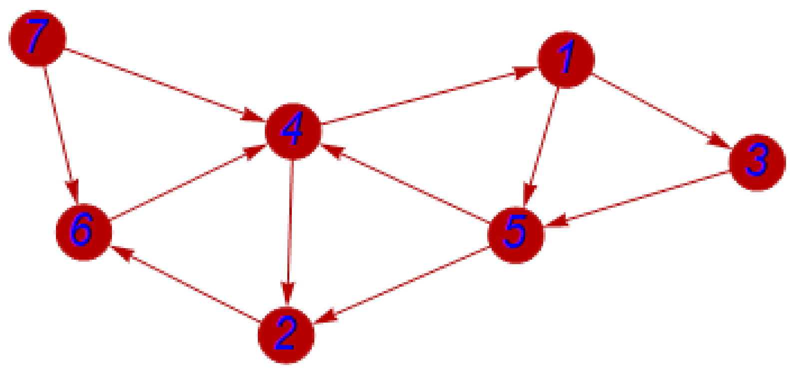


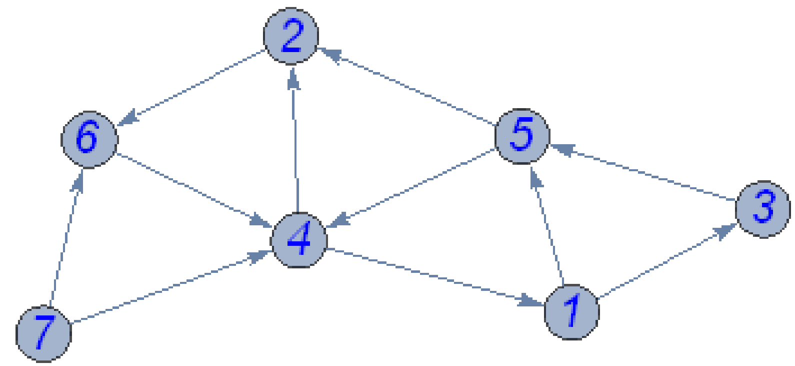






DFS





Notice: Though these graphs are not exactly the same, they are *isomorphic*.

How long does this take? $\sim |V| + |E|$

Applications: internet mapping, subroutine for *lots* of more advanced algorithms