ID1020 2018-10-01 #11

Undirected graphs

Discrete Math

- Set of vertices connected pairwise by edges
- Path - sequence of vertices connected by edges
- Cycle - first vertex = last vertex
- Connected
- Tree: acyclic
- Spanning tree
- Forest
- Spanning forest
- Bipartite graph

Sparse graph: $E \sim aV$ where a is small.

Graph API (simple)

Vertex representation: int 0—(V-1), symbol table

Graph(V) / Graph(in)
addEdge(v, w)
Iterable adj(v)
int V()
int E()

list of edges: linked list/array
adjacency matrix: (V by V) → adjacency lists; vertex indexed array of lists.

Real world graphs tend to be sparse

Depth-first search

Maze graph: Vertex = intersection, Edge = passage
Goal. Explore every intersection

Algorithm

- Unroll a ball of string behind you
- Mark each visited intersection and each visited passage
- Retrace steps when no unvisited options.

To visit a vertex v
- Mark v as visited
- Recursively visit all unmarked vertices adjacent to v.

Does not find the shortest path

Design pattern: Decouple graph data type from graph processing.

Paths API

Paths(Graph G, int s)
boolean hasPathTo(v)
Iterable pathTo(v)
<Integer>

DFS marks all vertices connected to s in time proportional to the sum of their degrees

After DFS, can check if vertex v is connected to s in constant time and can find v-s path (if exists) in ~

Application: flood-fill

Union find can be faster (because of caching)
Depth-first search: Constant time per query.

Breadth-first search
Put s into a FIFO queue, and mark s as visited
Repeat until queue is empty
- Remove vertex v from queue
- Add to queue all unmarked vertices adjacent to v and mark them.

Finds the shortest paths

Connectivity queries
v and w are connected if there is a path between them
Connected component: maximal set of connected vertices
Given connected components, we can answer connectivity queries in constant time.

Find connected component

Initialize all vertices v as unmarked.
For each unmarked vertex v, run DFS to identify all vertices discovered as part of the same component

Challenges:
Is a graph bipartite?
Finding cycles
Euler cycle
Hamilton cycle, NP-complete
Isomorphism, Open problem
Can a graph be represented plane, Difficult