Appendix B4 ✱

CSC263 Monday lecture : Graphs

Mar 5, 2018

- Graph $G = (V, E)$, V: set of vertices, E: set of edges
  - directed : $\langle u, v \rangle \neq \langle v, u \rangle$ // from a to b
  - undirected : $(u, v) == (v, u)$ // between a and b

- adjacency list : slot for every node, stores a linked list of the neighbours of that node (unordered)
  - directed : only store if node stores pointer to other node
    size = $n + m$
  - undirected : store neighbours in both, size = $n + 2m$
  - size $O(n+m)$, where $n = |V|$, $m = |E|$

- adjacency matrix : $n \times n$, stores 1 if edge between row/col nodes
  - to check if edge present, check slot either 1 or 0
  - ↳ checking for edge done in constant time
  - takes up more space, more efficient OPs than list

- Graph Exploration
  - starting node, traverse edges, discover new nodes
  - can discover : cycles, length of paths, etc.

- Breadth-First Search
  - colour[v] = W : undiscovered
    = G : discovered, unexplored
    = B : discovered, explored
    } vector storing state of nodes in process of exploration
  - parent[v] = u : "node v discovered v"
  - discovery path : $s \to v_1 \to v_2 - \dots - u - v$
  - D[v] = D[u] + 1 : length of discovery path

① first discovered, first explored : as you discover nodes, turn them grey and put them on the "to explore" FIFO queue

➢ BFS ALGO WALKTHROUGH

Problem Space
- $G=(V,E)$, $S \in V$

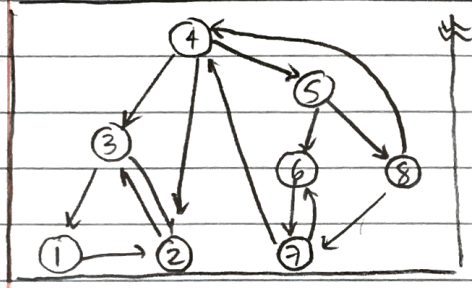
Setup
- BFS(G,S) :. $COLOUR(S) \leftarrow GREY$, $d[S] \leftarrow 0$, $P[S] \leftarrow NIL$, ENQ(Q,S)
  $\forall v \in V$, $COLOUR(v) \leftarrow WHITE$, $d[v] \leftarrow \infty$, $P[v] \leftarrow NIL$

Main loop
- WHILE Q IS NOT EMPTY:
  - $u \leftarrow$ DEQ(Q)
  - For each edge $(u,v) \in E$:
    - if colour(v) == WHITE, then:
      - colour$(v) \leftarrow$ GREY;
      - $D(v) \leftarrow D(u)+1$;
      - $P[v] \leftarrow u$
      - ENQ(Q,v)

      } "u discovered v"
  - $COLOUR(u) \leftarrow$ BLACK

➢ DIRECTED GRAPH EXAMPLE:

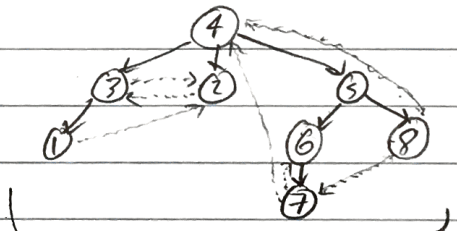
G:



1	2	3	4	5	6	7	8
2	3	1	3	6	7	6	7
		2	2	8		4	4
			5				

adjacency list

- BFS$(G,4)$ Q: 4_0
- $v=4$ Q: $3_1, 2_1, 5_1$
- $v=3$ Q: $2_1, 5_1, 1_2$
- $v=2$ Q: $5_1, 1_2$
- $v=5$ Q: $1_2, 6_2, 8_2$
- $v=1$ Q: $6_2, 8_2$
- $v=6$ Q: $8_2, 7_3$
- $v=8$ Q: 7_3
- $v=7$ Q:



d=0
d=1
d=2
d=3

[Reconstructed graph, BFS discovery tree]

Complexity of BFS $\in O(n+m)$
(traverse adjacency list)

- Claim: paths discovered via BFS are shortest possible path in graph to S. Prove in next class!