11. DFS

2018年11月28日 14:53

BFS

Graph search alg to find distance in graph
$BFS(G, s), G = (V, E), |V| = n, |E| = m, s$: starting point of G

Procedure
Initialization $(O(n))$
$init(G)$
    $for\ v\ in\ V - \{s\}$:
        $v.color\ =\ white$
        $v.distance\ =\ infty$
        $v.parent = Nil$
    $s.color = grey$
    $s.distance = 0$
    $s.parent = Nil$

The color means white: unexplored, grey: not fully, black: explored

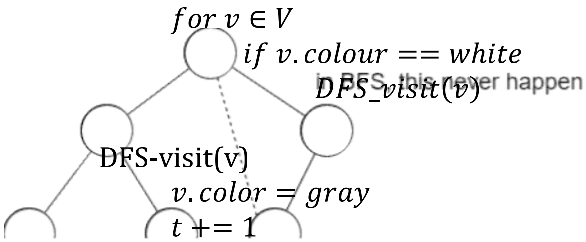
$BFS(G)\ (O(n+m))$
    $Q = \{s\}$ # $FIFO\ queue\ with\ only\ element\ s$
    $while\ Q \neq \emptyset$:
        $u = Dequeue(Q)$
        $for\ each\ neighbor\ v\ of\ u$:
            $if\ v.c = white$:
                $v.color = grey$
                $v.d = u.d + 1$
                $v.p = u$
                $enqueue(Q, v)$
        $u.c = black$

Define $\delta(s, u) =$ the distance of the shortest path, or ∞ if no such path
Theorem $\forall v \in V. \delta(s, u) = u.d$

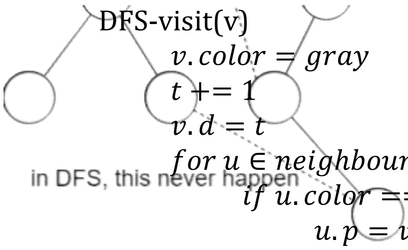
DFS

For each vertex v
    $v.color$ white | gray | black
    $v.d$ the discovery time of v
    $v.f$ the finish time of v

DFS(G)
    $t = 0$
    $for\ v \in V$
        $v.color = white$
    $for\ v \in V$
        $if\ v.colour == white$
            $DFS\_visit(v)$



DFS-visit(v)
    $v.color = gray$
    $t \mathrel{+}= 1$



DFS-visit(v)

    $v.color = gray$
    $t \mathrel{+}= 1$
    $v.d = t$
    $for\ u \in neighbours\ of\ v$
        $if\ u.color == white$
            $u.p = v$
            $DFS\_visit(u)$
    $v.color = black$
    $v.f = t$
    $t \mathrel{+}= 1$

Runs in $O(m+n)$

Properties

parenthetical property

$u.d < u.f < v.d < u.f \equiv u, v$ have no parenthetical relations

$u.d < v.d < v.f < u.f \equiv u$ is a ancester of v

proof Without losing generality, assume $u.d < v.d$

Case 1 since $u.f < v.d$, v is white when the other was discovered

Case 2 since $v.d < u.f$, u was gray when v was discovered.

white path property

v is a descendant of u IFF at time $u.d$, $\exists u \sim v$ be path of only white nodes

$\Rightarrow$ trivial

$\Leftarrow$ lemma $\forall w \in V. w\ on\ u \sim v \rightarrow w.f < u.f$

proof Let w' be the first vertex on the path s.t. $w'.f > u.f$. Then $w'.p$ on the path finished before exploring the edge $w' - p$. proven by well ordering

By parenthetical property, the claim is proven

Example G is an undirected graph. Running BFS on G at vertex u produces a tree T, running DFS on the u produces the same tree T. Then $G = T$

Topological Sort

Setup Let G directed, a topological order is a bijection $t: V \rightarrow [n]\ s.t. \forall e_i = (v_i, v_j), v_i \rightarrow v_j$ satisfy $t(v_i) > t(v_j)$

Theorem G has a topological order IFF it is acylic (no directed cycle)

$\Rightarrow$ consider the highest order vertex, it cannot have any edge pointing back, or it forms a cycle, contradiction

$\Leftarrow$ Proven in the algorithm below

Sink vertex with no out going edges

Algorithm 1 G is acyclic $\Rightarrow \exists$sink

$count = 1$

$while\ G \neq \emptyset$

$find\ a\ sink\ v$

$remove\ v\ and\ all\ its\ edges$

$make\ t(v) = count$

$count++$

Algorithm 2 Using DFS

run DFS and sort vertices by increasing $v.f$

lemma $e = u \to v, u.f > v.f$

proof Case 1 DFS visits u before v: $u \to v$ a white path at time $u.d$. By the white path theorem v is a descendant of u so $v.f < u.f$

Case 2 DFS visits v before u: Since the graph is acyclic, there does not exists $v \sim u$. Thus DFS on v must finish before even discovering u

Strongly Connected Components

Define $uRv \equiv \exists u \sim> v\ and\ v \sim> u$

Algorithm

Reverse all edges in G, call G_R

Run $DFS(G_R)$

Run $DFS(G)$ using the iterator of decreasing order of finish times in $DFS(G_R)$