Algorithms II

Ankur Kumar(14109) , Tushar Vatsal(14766)

Theoretical Assignment 3

1 A real life application of Directed Acyclic Graphs

Given a DAG G=(V, E) and s(root vertex) and t(exit vertex)
Pathid of a path p from s to t is defined as :

$$Pathid(p) = \sum_{e \in p} w(e), where\ e \in E$$

Problem Statement : To assign unique pathids to all possible paths from s to t such that the pathids range from 0 to $N-1$ where there are N possible paths from s to t.

Inductive Approach

Consider the topological ordering of given DAG(we shall refer to the vertices in this ordering only). Suppose for a vertex v, we assigned pathids to each path from source vertex 's' to any vertex w_i such that $(w_i, v) \in E$. Each path from s to w_i is assigned a unique pathid in the range $[0, m_i - 1]$ where m_i is the number of paths from s to w_i.
Let there are k vertices $\{w_1, w_2, w_3, ..., w_k\}$ such that $(w_i, v) \in E$. So total number of paths from s to $v = \sum_{i=1}^{k} m_i$. Then how to assign pathids from s to v such that we could assign unique pathids to $\sum_{i=1}^{k} m_i$ possible paths from 0 to $(\sum_{i=1}^{k} m_i - 1)$.
This can be done simply as mentioned below:
Notations: $\delta(i, j)$ = weight of edge (i,j) and P(i, j) = number of paths from i to j
Idea: Assign $\delta(w_1, v) = 0$, and $\forall\ i > 1$ assign $\delta(w_i, v) = \sum_{j=1}^{i-1} m_j = \sum_{j=1}^{i-2} m_j + m_{i-1} = \delta(w_{i-1}, v) + P(s, w_{i-1})$
Insight:
$s \rightsquigarrow w_1 \rightarrow v$ will give me all possible paths with pathids from 0 to $m_1 - 1$
$s \rightsquigarrow w_2 \rightarrow v$ will give me all possible paths with pathids from m_1 to $m_2 - 1 + m_1$
.
.
$s \rightsquigarrow w_i \rightarrow v$ will give me all possible paths with pathids from $\sum_{j=1}^{i-1} m_j$ to $m_i - 1 + \sum_{j=1}^{i-1} m_j$
.
.
$s \rightsquigarrow w_k \rightarrow v$ will give me all possible paths from $\sum_{j=1}^{k-1} m_j$ to $m_k - 1 + \sum_{j=1}^{k-1} m_j$

Thus we get all paths with pathids assigned from 0 to $(\sum_{i=1}^{k} m_i - 1)$
Thus we can get all paths with unique pathids assigned as asked in the question from s to t inductively.

Algorithm and Time Complexity

As mentioned above, wherever we will be writing k_{th} vertex, it means k_{th} vertex in topological ordering.
Step 1: We will first get the topological ordering of the given DAG $\triangleright$ **O(m+n)** time.
Step 2: Start processing from first vertex(source vertex, s) **iteratively** till we arrive at the exit vertex t..
Step 2(a): We have to assign every vertex number of paths coming from 1^{st} vertex to that vertex. Let P be an array where $P[v]$=number of paths from 1^{st} vertex to i^{th} vertex.
Assign $P[1] = 1$ and for any vertex $v > 1$, assign $P[v] = \sum_{k \in S} P[k]$, where $S = \{k | (k, i) \in E\}$ $\triangleright$ **O(in_degree(v))**.
Step 2(b): Assign weights to all edges (k, i) such that $(k, i) \in E$ in the way described in the inductive approach.
$\delta(w_1, v) = 0$, and $\forall\ i > 1$ $\delta(w_i, v) = \sum_{j=1}^{i-1} m_j = \delta + P[w_{i-1}]$ where $\delta = \delta(w_{i-1}, v)$
We would have calculated $\delta(w_{i-1}, v)$ and $P[w_{i-1}]$ beforehand. We will keep one variable δ and update it with $\delta(w_i, v)$ each time w_i is processed. O(1) time to assign weight to one edge. total number of edges = O(in_degree(v)). $\triangleright$ **O(in_degree(v))**
Clarification: Since we are assigning P values in topographical ordering, therefore assigning P values to any vertex i is not a problem because it requires P values of vertices j such that topographical ordering of j is before i. Also assigning weights to edges coming into any vertex i requires P values of vertices w such that $(w, i) \in E$ which we know beforehand since w comes before i in topographical ordering. Hence we scan the in-edges of all vertices in topographical ordering once.
Total Time = $O(m + n) + \sum_{i \in V} O(in\_degree(i)) = O(n + m)$

2 At least one path

Given graph $G = (V, E)$, to find if for all pair of vertices (u, v) in V either u is reachable from v or v is reachable from u. Find all the maximal SCCs(strongly connected components) of this graph G. This can be achieved in $O(m + n)$ time. Say there are k such maximal SCCs. Let S be the set of all such SCCs, where $S = \{s_1, s_2, s_3, ..., s_k\}$, where s_i is i^{th} maximal SCC. Treating each maximal SCC s_i as a vertex we get a graph say $G^{'} = (S, E^{'})$. This $G^{'}$ should be a DAG. Suppose $G^{'}$ is not a DAG implies there exist a cycle say : $(a_1 \rightsquigarrow a_2 \rightsquigarrow .... \rightsquigarrow a_i \rightsquigarrow a_1)$ where $a_j \in S \, \forall j \in \{1, 2, ..., i\}$
Just for the sake of clarity, I have shown edges between two SSCs though hook symbol. Take any two vertex u and $v \in V$ from this cycle.
Case1: Both u and v belong to same a_j
Since both the vertices u and v belong to an SCC, they are strongly connected to each other.
Case2: Vertex $u \in a_j$ and vertex $v \in a_k$, where $j < k$
Consider the path $u \rightsquigarrow a_{j+1} \rightsquigarrow .... \rightsquigarrow a_{k-1} \rightsquigarrow v$ It can be seen from this path that v is reachable from u. Also u is reachable from v : $v \rightsquigarrow a_{k+1} \rightsquigarrow a_i \rightsquigarrow a_1 \rightsquigarrow .... \rightsquigarrow a_{j-1} \rightsquigarrow u$
Note here u and v are vertices in V and a_js are SCCs. Thus any two vertices taken from SCC a_j where $j \in \{1, 2, ..., i\}$ are strongly connected which contradicts the fact that these SCCs are maximal. Hence graph $G^{'}$ is indeed a DAG. Since this is a DAG we can arrange S in topographical ordering. Let this ordering be $\{v_1, v_2, ..., v_k\}$ where $S = \{v_1, v_2, ..., v_k\} = \{s_1, s_2, s_3, ..., s_k\}$ but I am not claiming that $s_i = v_i$. Again this topological ordering can be done in $O(n + m)$ time in worst case when $k = n$.

Claim :

If there exists a path P: $v_1 \rightarrow v_2 \rightarrow .... \rightarrow v_k$ where $(v_i, v_{i+1}) \in$ edge in DAG of SCCs, then for all pair of vertices (u, v) in V either u is reachable from v or v is reachable from u in G otherwise not. This check can be achieved in $O(m + n)$ time(trivial).

Proof of Correctness of this claim

Suppose for some i, $(v_i, v_{i+1}) \notin E^{'}$. Then pick any vertex p from v_i and q from v_{i+1}. If $(v_i, v_j) \in E^{'}$, then $j \geq i + 2$ since j cannot be $i + 1$ and $j < i$ because in a topological ordering we cannot have any edge of the form (v_m, v_n), where $n < m$. So p is not reachable from q.
Also q is not reachable from p because $(v_i, v_{i+1}) \notin E^{'}$. If $(v_i, v_j) \in E^{'}$ then $j \geq i + 2$ and $\forall j \geq i + 2$ $(a_i, a_j) \notin E^{'}$ from topological ordering. Thus q is also not reachable from p.
Now suppose there exists such a path then for all pair of vertices (u, v) in V either u is reachable from v or v is reachable from u.
Choose any two vertices p and q.
Case1: p and q belong to same v_i
Then there is a path from p to q and from q to p since they belong to an SCC and hence they are strongly connected.
Case2: $p \in v_j$ and $q \in v_k$ say $j < k$
In this case consider a path from p to q which is shown below :
$p \rightsquigarrow v_{j+1} \rightsquigarrow .... \rightsquigarrow v_{k-1} \rightsquigarrow q$.
Thus there exists a path from p to q Hence proved. Thus my claim is correct.