Distance Sensitivity Oracle for $W \times W$

Problem 1 (1-FT-distance oracle)**.** *Let $G = (V, E)$ be a undirected graph, and W be a subset of V. The goal is to preprocess G to compute a compact data-structure $\mathcal{D}$ such that given any pair of vertices $(s, t) \in W \times W$ and any edge $e \in E$, the data-structure $\mathcal{D}$ can report "s to v distance in $G - e$" in $O(1)$ time.*

1 Construction of $O(n|W| \log n)$ sized oracle

As an input we are given a set W of vertices in an **undirected unweighted** graph $G = (V, E)$. Our data structure stores the following information:

Information about shortest-path-trees rooted at nodes in W: For each vertex $w \in W$, we store

1. A shortest path tree rooted at w, i.e. trees T_w.
2. The pre-order and post-order numbering of vertices in trees T_w, for $w \in W$
3. The depth of each node v in trees T_w, for $w \in W$.
4. A level-ancestor data-structure for trees T_w, for $w \in W$.
5. The distances $d_G(w, v)$, for each $v \in V$ and $w \in W$.

The total space used for above information will be $O(n|W|)$. See Figure 1.

Information on specific EDGE failure on shortest-path between pairs in set $(W \times V) \cup (V \times W)$: For every pair $(s, t) \in (W \times V) \cup (V \times W)$ and every integer $0 \leq i \leq \log_2 d_G(s, t)$, we store

$$B1(s, t, i) = d_{G-e}(s, t),$$

where $e = (u, v)$ is an edge lying on $\pi_G(s, t)$ and satisfying $d_G(s, u) = 2^i$. See Figure 2.

Information on specific SEGMENT failure on shortest-path between pairs in set $(W \times W)$: For every pair $(s, t) \in (W \times W)$ and every integer $0 \leq i \leq \log_2 d_G(s, t)$, we store:

$$B2(s, t, i) = d_{G-edges\left(\pi(u,v)\right)}(s, t),$$

where u, v are vertices on $\pi(s, t)$ satisfying (i) $d_G(s, u) = 2^i$, and (ii) $d_G(u, v) = 2^i$. See Figure 3.

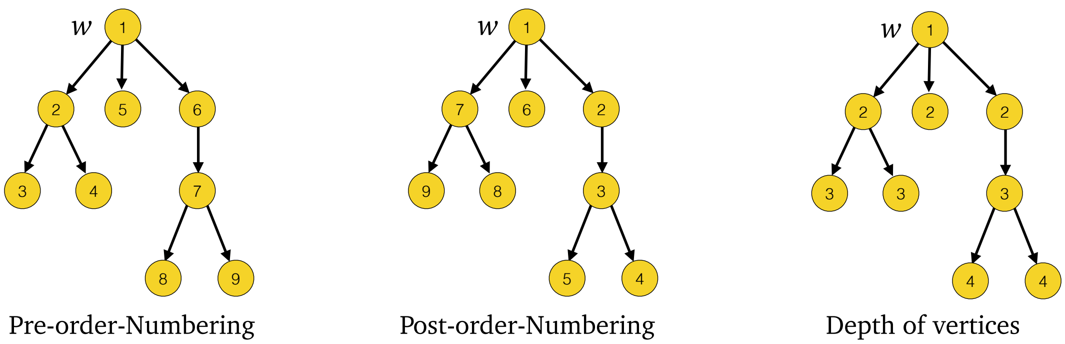

Figure 1: (i) Depiction of Pre-order-numbering, Post-order-numbering, and depth of vertices in a tree. (ii) A vertex x is ancestor of a vertex y in T if and only if Pre-order$_T(x)$ <Pre-order$_T(y)$ and Post-order$_T(x)$ <Post-order$_T(y)$

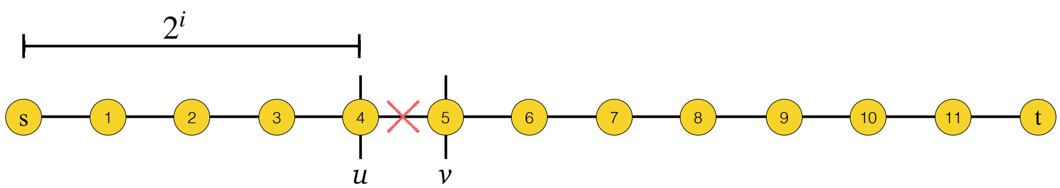

Figure 2: The vertices u, v in structure $B1(s, t, i)$ under $i = 2$.

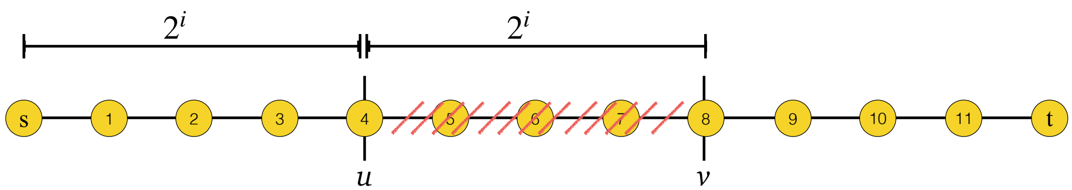

Figure 3: The vertices u, v in structure $B2(s, t, i)$ under $i = 2$.

2 Query Algorithm

Let $(s, t) \in W \times W$ be a query pair and $e = (u, v)$ be a failing edge lying on $P = \pi_G(s, t)$.

Whether e lies on $\pi_G(s, t)$ or not can be verified in $O(1)$ time using pre-order and post-order numbering, and depth of vertices in T_w, for $w \in W$).

Let i_0 and j_0 be greatest integers satisfying $2^{i_0} \leq d_G(s, u)$ and $2^{j_0} \leq d_G(v, t)$. Let s', t' be vertices on $\pi(s, t)$ such that $d_G(s', u) = 2^{i_0}$ and $d_G(v, t') = 2^{j_0}$. (See Figure 4). The vertices s', t' can be computed in $O(1)$ time by using the Level-Ancestor data-structure on shortest path trees T_w, for $w \in W$.

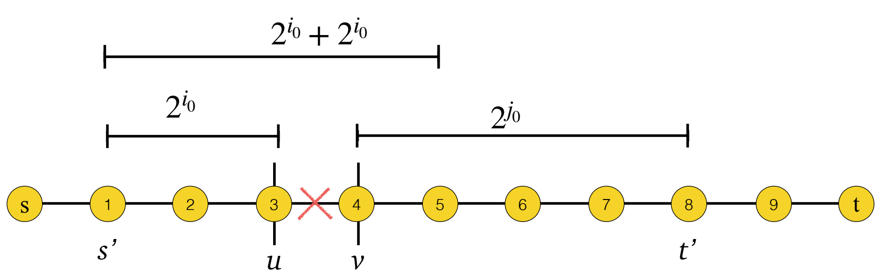

Figure 4: The vertices s', t' on path $P = \pi_G(s, t)$ in G. (The figure considers the scenario when $i_0 \leq j_0$).

We have the following two cases.

Case 1: There is an s to t shortest path in $G - e$ passing through s' or t'

(i) The path $\pi_{G-e}(s,t)$ passes through t', then $d_{G-e}(s,t) = d_{G-e}(s,t') + d_G(t',t)$.
The distance between e (i.e. endpoint v) and t' is 2^{j_0}, so $d_{G-e}(s,t')$ can be retrieved from $B1$.

(ii) The path $\pi_{G-e}(s,t)$ passes through s', then $d_{G-e}(s,t) = d_G(s,s') + d_{G-e}(s',t)$.
The distance between e (i.e. endpoint u) and s' is 2^{i_0}, so again $d_{G-e}(s',t)$ can be retrieved from $B1$.

Case 2: All s to t shortest path in $G - e$ avoids the entire segment $P[s', t']$

Let us assume that $i_0 \leq j_0$. (If $j_0 < i_0$ then a similar analysis will follow).

Let u_0, v_0 be vertices on $P = \pi_G(s,t)$ such that $d_G(s,u_0) = d_G(u_0,v_0) = 2^{i_0}$. Since $2^{i_0} \leq 2^{j_0}$, we have:

$$s' \leq_P u_0 \leq_P u \textit{ and } v \leq_P v_0 \leq_P t' .$$

Thus replacement path $\pi_{G-e}(s,t)$ does not pass through segment $P[u_0, v_0]$, so, $\pi_{G-edges\left(P[u_0,v_0]\right)}(s,t)$ is an $s - t$ shortest path in $G - e$. Thus in this case, we can use B2 to report $d_{G-e}(s,t)$.

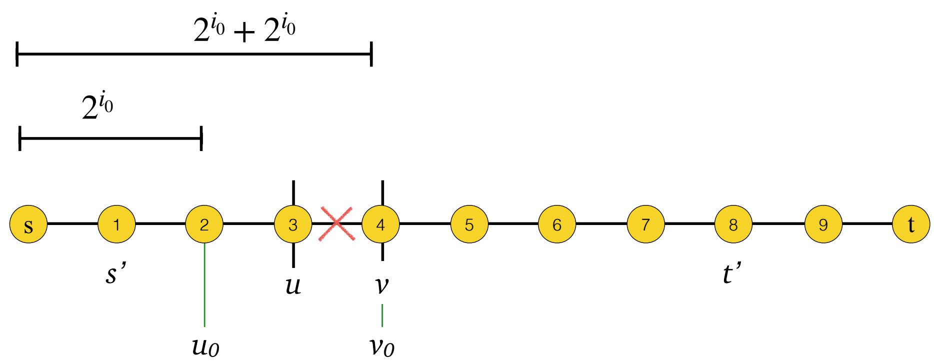

Figure 5: The vertices u_0, v_0 on path $P = \pi_G(s,t)$ in G. (The figure considers the scenario when $i_0 \leq j_0$).

Remark For the sake of simplicity we only discussed the edge-failure case in unweighted undirected graphs, but the results naturally extend to the vertex failures as well as weighted directed graphs.