Przemysław Uznański
Chih-Hung Liu

April 11, 2017

Advanced Data Structures

Spring Semester 2017

Exercise Set 8

Exercise 1:

Given a dynamic *weighted* graph $G(V, E)$, where $|V| = n$ and $|E| = m$, we attempt to extend the fully dynamic connectivity data structure to build a dynamic *minimum spanning tree* data structure that allows only deletion operations. Recall the key ingredients for the fully dynamic connectivity as follows

- Assign each edge a *level* that starts at $\log n$ but only decreases over time.
- Let G_i be the subgraph of G consisting of edges with level at most i, i.e., $G_0 \subseteq G_1 \subseteq \cdots \subseteq G_{\log n} = G$.
- Let F_i be a spanning forest of G_i for $1 \leq i \leq \log n$, and let F be $F_{\log n}$.

Two invariants are required for fully dynamic connectivity:

Invariant 1. Every connected component G_i has at most 2^i vertices.

Invariant 2. $F_0 \subseteq F_1 \subseteq F_2 \subseteq \cdots \subseteq F_{\log n}$, and F is a minimum spanning tree/forest of G if the level of an edge is interpreted as its weight.

If the deletion of an edge $e = (u, v)$ separates a tree $T \in F$ into two subtree T_v and T_u, then we need to find the lightest edge connecting T_v and T_u as the replacement edge. Therefore, another invariant is suggested:

Invariant 3. Every cycle C has a non-tree edge of maximum weight and maximum level among all the edges in C.

Please complete the following two tasks:

- Prove that among all the replacement edges, the lightest edge is on the minimum level.
- Assume the level of e to be ℓ, and describe how to find the replacement edge.

Hint: Consider two replacement edge e_1 and e_2 where the weight of e_1 is larger than the weight e_2. Before the deletion of e, inserting e_1 (resp. e_2) into F will form a cycle C_1 (resp. C_2). Compare the levels of e_1 and e_2 using Invariant 3 and the cycle $C = C_1 \cup C_2 \setminus C_1 \cap C_2$.

Exercise 2:

Please prove the lower bound of a deletion operation for the dynamic minimum spanning tree data structure to be $\Omega(n)$.

Hint:

- Reduce the standard sorting problem to a sequence of deletion operations.

Exercise 3:

In Exercise 3 of list 6 there was one missing ingredient: how to maintain information on which line segments border which faces, under segments deletion/insertion. Show that it can be maintained in $\mathcal{O}(\log n)$ per insertion/deletion/query, using ideas from Euler-tour trees.