Dynamic bounded out-degree orientations

Eva Rotenberg

References and reading

Dynamic Representations of Sparse Graphs, Gerth St0lting Brodal and Rolf Fagerberg, 1999

Exercises

1 Prerequisites

1.1 Give a data structure for adjacency that has $O(m)$ space and answers adjacency in $O(\log n)$ time. How fast can you update your adjacency data structure? Can you get $O(\log n)$?

1.2 For a static, non-changing graph. Give a data structure that answers adjacency queries in $O(1)$ worst-case time and takes $O(n+m)$ space. Can you construct it in linear expected time?

2 Arboricity and out-degrees

2.1 Show how to out-orient a forest, such that each vertex has at most 1 out-edge.

2.2 Give an example of a graph which can be out-oriented such that each vertex has at most 1 out-edge, but which needs 2 forests to cover its edges.
(I.e. it is not a forest, but the edge set is the union of 2 forests.)

2.3 [$*$] (i) Show that a graph with a Δ-orientation has a vertex of degree $\leq 2\Delta$, (ii) use this to divide its edges into $O(\Delta)$ forests.

3 Simplification

We study the statement of Lemma 2 in the setting where $\delta = 2c$.

3.1 Let G be a graph of arboricity $\leq c$. If vertex u has out-degree $2c$, show that there is a vertex v of degree $< 2c$ and a directed path u to v of length $\leq \log_2 n$.

4 Dynamic forest orientation

In this section, we consider the question of dynamically orienting a forest with worst-case update time.

4.1 Consider a dynamic forest of paths. Consider the task of giving it a 1-orientation. Give an example of an update sequence where $O(n)$ vertices need to change direction.

4.2 Show that one can efficiently 2-orient a dynamic forest of paths.

4.3 Show that one can 2-orient a dynamic forest with $O(\log n)$ worst-case time per insertion or deletion.
(Hint: path decompositions.)

5 Acyclic Δ-orientations [$*$]

[$*$] Consider now the case where you want a Δ-orientation that is furthermore *acyclic*, so that there are no directed cycles in the oriented graph. Modify the algorithm (and the analysis) to accommodate this.