Weekplan: Dynamic Graphs

Philip Bille   Inge Li Gørtz   Eva Rotenberg

References and Reading

[1] Dynamic Representations of Sparse Graphs, Gerth Brodal and Rolf Fagerberg, 1999

Exercises

1 Space Travel

The purpose of this exercise is to train modelling and problem solving for dynamic graphs. (Not as-such relating to the material.)

1.1 You are commanding the space ship Traveller stuck in a distant part of the galaxy. On board, you have n members of the crew. You now want to divide this crew into four shift teams, 1, 2, 3, 4, that work on the space ship at disjoint times of the day. Challenge: you want to avoid putting a person on the same team as one of their arch enemies. Each team should contain at least 1 person and at most $n-3$ persons, and n is greater than 4. Luckily, due to their advanced star fleet training, each member of your crew is in a conflict with at most 3 different people of the star ship at the same time.

- Show how to model this as a graph problem, and,
- Give an efficient algorithm for solving it.
- What is the running time of your algorithm?

1.2 Now, as time goes by, relationships change, enemies become friends, and vice versa. Still, at all points of time, no person is in a conflict with more than 3 people.

- Show how to model this as a dynamic graph.
- How fast can you perform the necessary update when a conflict disappears?
- How fast can you perform the necessary update when a new conflict arises?
- Is your solution better than assigning people to new teams all over again?

Definition (colouring) A proper k-colouring (or just colouring) of a graph is an assignment of colours $1, 2, \ldots, k$ to the vertices of the graph, such that no nieghbours have the same colour. (In other words, no edge is monochromatic).

2 Prerequisites

2.1 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 $O(\log n)$?

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

3 Arboricity and out-degrees

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

3.2 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.)

3.3 (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.

4 Simplification (Also covered in lecture.)
We study the statement of Lemma 2 in the setting where $\delta = 2c$.

4.1 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 $\leq \log_2 n$.

5 Adaptivity In this exercise, we consider the case where the arboricity of the graph also changes.

5.1 Show how to extend the structure so that you at all times have $O(c_{\max})$ out-edges, where $c_{\max}$ is the maximal arboricity of the dynamic graph thus far.

6 Acyclic Δ-orientations. [$*$]

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