Data Structures and Algorithms Spring 2024 — Problem Sets

by Nikolai Kudasov

March 25, 2024

Week 10. Problem set

1. Assume that vertex labels in a graph $\mathcal{G}$ are integers. Imagine a variation of the adjacency-list graph representation [Goodrich, §14.2.2], where instead of lists we use Red-Black trees [Cormen, §13] with vertex labels serving as keys in the search tree. What are the worst-case time complexities of the following Graph ADT operations over this modified representation?

   (a) `insertVertex(v)`

   (b) `insertEdge(from, to, e)`

   (c) `removeEdge(from, to)`

   (d) `removeVertex(v)`

   Assume that `v`, `from`, and `to` arguments above are *labels*, not references to vertex objects.

2. Would you use the adjacency matrix structure [Goodrich, §14.2.3] or the adjacency list structure [Goodrich, §14.2.2] in each of the following cases? Justify your choice in each case.

   (a) The graph has 20,000 vertices and 80,000 edges, and it is important to use as little space as possible.

   (b) The graph has 20,000 vertices and 80,000,000 edges, and it is important to use as little space as possible.

   (c) The graph has 20,000 vertices and 20 edges, and you need to answer the query `getEdge(u, v)` as fast as possible, no matter how much space you use.

   (d) The graph has 20,000 vertices and 80,000 edges, and you need to answer the query `getEdge(u, v)` as fast as possible, no matter how much space you use.

3. In your own words, explain why, for an adjacency-list representation of a graph [Goodrich, §14.2.2], the method `removeVertex(v)` takes $O(\mathsf{deg}(v))$ time (worst case) and does not depend on the number of vertices in a graph. You *must* explicitly specify what contributes to the running time of the method, as well as explicitly name the data structures involved that make the efficient implementation possible.

References

[Cormen] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein. *Introduction to Algorithms, Fourth Edition.* The MIT Press 2022

[Goodrich] M. T. Goodrich, R. Tamassia, and M. H. Goldwasser. *Data Structures and Algorithms in Java.* WILEY 2014.