Weekplan: Streaming II.

Philip Bille   Inge Li Gørtz   Christian Wulff-Nilsen

References and Reading

[1] Amit Chakrabarti: *Data Stream Algorithms* 2011 (updated July 2020) Chapter 14.

Exercises

1 Warmup How much space in terms of n do you need in the worst case do describe a graph with n vertices?

2 Connectivity

2.1 Design a semi-streaming algorithm that counts the number of connected components of a graph. Analyze the space consumption, update time, and query time of your algorithm.

2.2 Show that the Graph Connectivity Algorithm in the slides maintains F as a spanning forest of the part of G seen so far in the stream.

3 Bipartiteness The following exercises fill in the missing parts related to bipartiteness testing in the slides.

3.1 Show that a graph is bipartite if and only if it is 2-colorable.

3.2 Show that any forest has a 2-coloring.

3.3 Argue that in one of the directions in the correctness proof for the Bipartiteness Testing Algorithm, the path P_{uv} indeed exists.

4 Spanners The following exercises refer to definitions in the slides.

4.1 Show that H is a t-spanner of G if and only if H has the t-spanner edge property (the proof is in the slides so try to avoid looking at it).

4.2 Show that the graph H satisfies $\gamma(H) \geq t + 2$ at any point in the t-spanner algorithm.

5 Matching A *matching* in a graph $G = (V, E)$ is a set $E' \subseteq E$ such that no two edges in E' share an endpoint. A *maximal matching* is a matching with the property that no additional edge can be added to it while keeping it a matching. A *maximum cardinality matching* is a matching of maximum size.

5.1 Give an example of a graph containing a maximal matching of size smaller than a maximum matching.

5.2 Give a semi-streaming algorithm that computes a maximal matching of a graph. Show its correctness and analyze its space consumption and update time.

5.3 Give a semi-streaming algorithm that computes a 2-approximate maximum cardinality matching M of a graph, i.e., M should be a matching satisfying $|M| \geq \frac{1}{2}|M^*|$ where M^* is a maximum cardinality matching. Show the correctness of the algorithm and analyze its space consumption and update time.

6 k**-center clustering** In this exercise, we focus on a problem related to graph streaming. We are given a metric space (X, d) and a positive integer k. For a subset $Y \subseteq X$ and an $x \in X$, define $d(Y, x) = d(x, Y) = \min_{y \in Y} d(x, y)$; if $Y = \emptyset$, define $d(Y, x) = d(x, Y) = \infty$.

Instead of a stream of edges, we get a stream of points from X: $x_1, x_2, \ldots$. Let S denote the set of points from the stream. The output should be a set of *centers* $Y \subseteq S$, $|Y| \leq k$, such that the cost

$$\max_i d(x_i, Y) = \max_{x \in S} d(x, Y)$$

is minimized.

Let OPT denote the cost an optimal solution. We will analyze a semi-streaming algorithm using $O(k)$ space which achieves a solution of cost at most $r = 2\text{OPT}$.

The algorithm works as follows. Initialize $Y \leftarrow \emptyset$. For the currently processed x_i, check if $d(x_i, Y) > r$ and if so, add x_i to Y; otherwise, do nothing.

A big downside of the algorithm is that it needs to know r. Fortunately, there is a modification of the algorithm that does not have this requirement and which obtains a solution of cost at most $(2+\epsilon)\text{OPT}$ for any chosen constant $\epsilon > 0$. We will not focus on this.

6.1 Show that the algorithm finds a set of at most k clusters.

6.2 Show that the cost of this solution is at most r.

7 Girth theorem As part of the proof of the girth theorem in the slides, show that B_u is a tree (try to avoid looking at the solution in the slides).