ADFOCS Exercise Set #1

Danupon Nanongkai

Main Problems

1. **Weighted Global Min-Cut.** Given an undirected, weighted, graph $G = (V, E)$, a global min-cut is a partition of V into two subsets (A, B) such that the sum of weights of edges between A and B is minimized. Prove that maintaining the value of global min-cut exactly under the following operations admits no $O(n^{1-\epsilon})$ amortized update time assuming the OMv conjecture:

   - **Initialize(n):** Create an empty n-node graph.
   - **Insert(u, v, w):** Insert an edge between nodes u and v of weight w, if such edge does not already exist.
   - **Delete(u, v):** Delete edge (u,v)

Related works**:** In contrast to the above, it was known that one can maintain a $(1+\epsilon)$-approximate value of global min-cut in $O(\sqrt{n})$ time. It is a major open problem whether this can be improved to $O(polylog\ n)$ (such update time exists for $(2+\epsilon)$-approximation).

2. **Perfect matching.** Given an undirected, unweighted, graph $G = (V, E)$, a matching is a set of edges without common vertices. The perfect matching is a matching which matches all vertices of the graph. Prove that maintaining if the graph has a perfect matching under edge insertions and deletions admits no $O(n^{1-\epsilon})$ amortized update time, assuming the OMv conjecture.

Remark: If you find the above too hard, try to prove a lower bound for maximum matching instead.

3. **Matching without augmenting paths of length 5.** An augmenting path for a matching M is a path with an odd number of edges $e_1, e_2, \dots, e_k$ such that $e_{odd} \notin M$ not in M and $e_{even} \in M$. Consider the problem of maintaining a matching without an augmenting path of length 5 or less, where after each edge deletion and insertion the algorithm has to output how the maintained matching changes. Prove that an algorithm for this problem admits no $O(n^{1-\epsilon})$ amortized update time, assuming the OMv conjecture.

Related works**:** Since perfect matching admits a high lower bound, recent research has been on *approximating* maximum matching size. The 2- and the 3/2-approximation algorithms of Baswana et al. (FOCS'11) and Neiman-Solomon (STOC'13) exclude length 1 and 3 augmenting paths. The above shows a huge lower bound for the same approach for 5/4-approximation.

4. **(Open-ended question) Dynamic diameter.** Prove as high lower bound as possible for maintaining the diameter of an unweighted graph undergoing edge insertions and deletions.

Remark: Don't be surprised if the OMv conjecture does not imply a strong lower bound.

5. **(Bonus question by Jan van den Brand) Matrix inverse under row and column updates.** Consider the problem of maintaining a matrix inverse (over finite fields or rational numbers). An algorithm for this problem should handle the following operations:

   - **Initialize($\boldsymbol{n}, \boldsymbol{i}, \boldsymbol{j}$):** Create an $n \times n$ identity matrix $\boldsymbol{A}$. Fix the value of i and j (the value of A_{ij}^{-1} has to be returned after every update).
   - **Row-Update ($\boldsymbol{k}, \boldsymbol{v}$):** Change the k-th row of A to vector v.
   - **Column-Update ($\boldsymbol{k}, \boldsymbol{v}$):** Change the k-th column of A to vector v.

   After each update, the algorithm should output the value of A_{ij}^{-1} or output that A is not invertible. Prove that an algorithm for this problem admits no $O(n^{2-\epsilon})$ amortized update time, assuming the OMv conjecture.

Related works: In contrast to the above, $O(n^{2-\epsilon})$ worst-case update time can be achieved if only row- or column-updates are allowed [Sankowski, FOCS'04].

Other problems (to warm-up and complete gaps from the lectures)

a) A vertex cover in a graph is a set of nodes S such that for every edge $(u,\ v)$, either u or v is in S. Consider the problem where a fixed graph G is given and an update is an insertion or deletion of a node to and from S. After each update, the algorithm has to say whether S is a vertex cover. Prove that this problem admits no $n^{1-\epsilon}$ amortized update time.

b) In the lecture, we proved that there is no dynamic st-reachability algorithm with $n^{1-\epsilon}$ amortized update time. Show that there is also no algorithm with $m^{\frac{1}{2}-\epsilon}$ amortized update time.

c) In the lecture, we sketched how to reduce from the OMv conjecture to the $OuMv$ conjecture. Show a reduction from the OMv conjecture to the $\gamma - OuMv$ conjecture.