# Approximating distances in graphs
Uri Zwick

## Review exercises

1. Show that with high probability, all distances larger than $2n^{1/2} \ln n$ are correctly computed by the $O(n^{5/2})$ time surplus-2 algorithm.

2. Show that the following simple algorithm constructs, in $O(m)$ time, a $(2k-1)$-spanner with at most $n^{1+1/k}$ edges of any undirected unweighted graph:

   While the graph is not empty, choose a vertex $v \in V$.

   Let $N_i(v)$ be the set of vertices of distance at most $i$ from $v$. Let $j$ be the smallest integer for which $|N_j(v)| \leq n^{1/k}|N_{j-1}(v)|$.

   Add to the spanner a tree of shortest path from $v$ to all vertices of $N_j(v)$ and delete the vertices of $N_{j-1}(v)$ and all the edges adjacent to them from the graph.

3. The girth of a graph is the size of the shortest cycle in the graph. Use the construction of the previous exercise to show that the girth of any graph with at least $n^{1+1/k}$ edges is at most $2k$.

4. Let $m_g(n)$ be the maximum number of edges in an n-vertex graph of girth at least $g$. Show that every $n$-vertex graph has a $t$-spanner with at most $m_{t+2}(n)$ edges and that this result is best possible.

5. Complete the proof that the algorithm of Baswana-Sen algorithm produces a $(2k-1)$-spanner with at most $O(kn^{1+1/k})$ edges. Show that it can be implemented to run in $O(km)$ time.

6. Prove the correctness of the following variant of the Baswana-Sen algorithm: Perform only the first $(k-1)/2$ iterations of the original algorithm. (Assume that $k$ is odd.) For every pair of trees produced in the last iteration, find the lightest surviving edge that connects them and add it to the spanner.

7. What is the maximum stretch of the variant of the query answering algorithm of Thorup and Zwick that does not swap $u$ and $v$ in each iteration, i.e., finds the smallest $i$ for which $w = p_i(u) \in B(v)$ and returns $\delta(u, w) + \delta(w, v)$ ?

8. Show that the subgraph composed of the shortest path trees of the clusters constructed by the Thorup-Zwick approximate distance oracles data structure is a $(2k-1)$-spanner with an expected number of $O(kn^{1+1/k})$ edges.