Columbia University
in the city of new york
Computer Science

Prof. Alexandr Andoni
Department of Computer Science
Columbia University
New York, NY 10027, USA
212-853-0685
andoni@cs.columbia.edu

Re: PhD Thesis of Wojciech Janczewski

This is an enthusiastic endorsement of the PhD candidate Wojciech Janczewski's PhD thesis entitled "Graph Labeling Schemes and Dynamic Longest Increasing Subsequence". The thesis is based on a few papers published in top conferences in Theretical Computer Science/Algorithms field, such as STOC and SODA. Overall, the thesis is of highest quality, at the international level, and strongly merits awarding PhD (e.g., it would be absolutely supported at my own institution). Below I will specify some detailed merits of the thesis.

The thesis is composed of two parts (as suggested by the title), both of core interest in the Theoretical Computer Science community. The first one is on distance labelling schemes, which is primarily an information-theoretic question but whose solution is typically designed via explicit algorithns. The second one is on dynamic algorithms for the Longest Increasing Sequence, a particularly classic algorithm design question. Both contributions are excellent contributions to the field of algorithm design. Below I describe in detail each of these directions.

Graph labelling schemes are a fundamental algorithmic primitive for *de*-centralizing information in a graph/network. In particular, for a given graph, a labelling scheme assigns a short label to each node so that standard graph notions can be computed from these labels only: notions such as connectivity, distance, routing information, and many others. For example, in the case of the adjacency problem, given labels of two nodes in the graph, one would like to be able to determine whether the nodes are adjacent or not in the graph. These problems have emerged at least as early as 1992, with many deep connections to both theory and practice. For the latter, consider the problem of routing a packet in a distributed network: a (routing) labelling scheme would allow a packet carry a short label that's sufficient for routing the packet to its eventual destination (while each node in the network stores a small amount of information, much smaller than the description of the entire network). On the theory side, some labelling schemes (notably for adjencency) are tightly connected to combinatorial notions such as universal graphs, objects otherwise studied by mathematicians.

The thesis has three contributions in the area of graph labelling schemes. The first one is for routing in trees. While trees are a simple topology for a graph, trees are nonetheless core to understanding the more general cases (and indeed, some algorithms reduce to routing on trees). In this setting, the thesis improves the label size to $\log n + O(\log \log n)^2$ from earlier $> \log n + \log n / \log \log n$, ie exponential improvement in the second term. Note that, when talking about label sizes — like in coding — the constants matter (think of the label as the packet header for which we want a minimal number of bits). Furthermore, the size of the label determines the size of the related

universal graph, whose size is exponential in the label size and hence even "small" improvements in the label size entail large improvements in the universal graph size. As a conclusion, improvements in the second term are a significant contribution.

The second considered graph labelling problem is that of determining adjacency in a planar graph, for which the thesis shows a significantly simpler algorithm, obtaining labels of size $\log n + O(\sqrt{\log n})$. Finally, the third contribution is a labelling scheme for computing the distance in the permutation graphs. Notably, the presented algorithm obtains a tight first term, i.e. a label of size $3 \log n + o(\log n)$, mathcing a lower bound that has previously shown for this case.

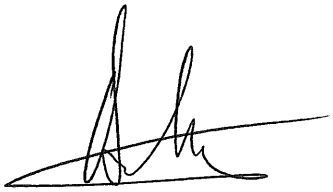
Longest Increasing Sequence (LIS): let me now focus on the second part, on algorithms for LIS. Finding the LIS in a sequence is a very classic algorithmic question, for which an algorithm was already designed in '70s, and that has been very influential and recurring in Theoretical Computer Science. For example, beyond the obvious many applications, LIS has often been one of the founding, driving problems in new models, such as streaming, sketching, property testing — and as such, had had a lot of impact. Perhaps quite surprisingly, the *dynamic* version of the problem hasn't been studied until very recently: this is the version where the sequence is dynamic (allows insertions and deletions), and we need to maintain a data structure that allows fast operations to update the sequence and query the length of the current LIS. Given that the classic LIS problem is solvable in time $O(n \log n)$ for length-n sequences, it is reasonable to desire a solution that has $\text{poly}(\log n)$ time per operation.

This thesis was the first to design such a solution for dynamic LIS problem when allowing for a $1+\epsilon$ approximation, for arbitrary small $\epsilon > 0$. In particular, it achieves update and query times of $\text{poly}(\frac{\log n}{\epsilon})$, a major improvement over the state-of-the-art. Prior work showed that one can obtain $(1/\epsilon)^{O(1/\epsilon)}$-factor approximation with update/query time of n^{ϵ} time; this means that even for a (large) constant approximation, one would need polynomial query time. Hence the contribution here not only improves the approximation to a much more palatable $1+\epsilon$ approximation, but also improves the runtime exponentially (and is likely now only polynomially off from the best possible algorithm).

As one would expect from such a basic problem, the LIS algorithm from this thesis has already had impact in the field, including in my own paper from FOCS'22. In particular, the algorithm from this thesis can inf fact handle a larger range of operations: e.g., it is able to report the actual LIS (not just the length) in time close to linear in the length of the LIS (which is close to the best possible time). We used crucially this precise algorithm in our paper to obtain the best known *sublinear-time* algorithm for LIS, which itself was used to obtain the best approximation for the related problem of Longest Common Subsequence (LCS) in linear time. LCS is the most common way to compute similarity between two sequences such as genes, and indeed has numerous applications in Computational Biology. Computing LCS requires quadratic time (under standard computational conjectures), and hence obtaining fast approximation is has been a goal for quite a while. To restate: the algorithm from this thesis is a core component of the best approximation algorithm for LCS today.

In addition to upper bounds, the thesis also includes lower bounds on the dynamic LIS problem, at least in part justifying the focus on $1+\epsilon$ approximation (as opposed to exact algorithms). Indeed, the thesis shows a polynomial lower bound on query time for the operation of exactly reporting the LIS of a contiguous subsequence (the algorithm from the thesis solves this type of operation with $1+\epsilon$ approximation efficiently). Note that the best algorithms for exact LIS, even for the entire string, remain polynomial time.

Overall, the contributions in this thesis are of the highest quality. The thesis is also well-written and will likely serve as a nice survey on the topics covered.



Alexandr Andoni
Associate Professor, Department of Computer Science
Columbia University in the City of New York