

## Papers for writing an essay in Advanced Algorithmics 2013 Spring class

Choose one of the below articles/topics and write a concise, 2-page essay on the paper focusing on algorithmic ideas presented or used in the papers. Follow traditional article formatting rules of a conference. E.g. use LNCS TeX style. Don't forget to include proper references. I recommend a 2-column layout. Essay – **2 pages long (exactly)**. No single direct quotations allowed. Use your own wording. Or cite with quotes, if absolutely necessary. Include 2-3 core other citations that would help open up the article, too.

Springer/LNCS styles: <http://www.springer.com/computer/lncs?SGWID=0-164-7-72376-0>

AAAI style: <http://www.aaai.org/Publications/Author/author.php>

Deadline: 17<sup>th</sup> of May (before 24h midnight). Every delayed date (EET time, even 1 minute into May 18th) will reduce points by 25% of maximum. By Tuesday there is no point submitting anymore. Submit a PDF file.

- **Bitcoin – what is the exact mining challenge and how much can I earn with my current CPU-year?** (an open question topic proposed by JV)
- **Fast Exact Shortest-Path Distance Queries on Large Networks by Pruned Landmark Labeling** Takuya Akiba, Yoichi Iwata, Yuichi Yoshida  
<http://arxiv.org/abs/1304.4661>
- **Sublinear time algorithm for PageRank computations**, Christian Borgs, Michael Brautbar, Jennifer Chayes, and Shang-Hua Teng, [arXiv:1202.2771](https://arxiv.org/abs/1202.2771) and WAW 2012. It's only slightly sublinear but it's interesting to me that we can prove such bounds, in a natural model of computation for web graph access.
- **Finding a most biased coin with fewest flips**, Karthekeyan Chandrasekaran and Richard Karp, [arXiv:1202.3639](https://arxiv.org/abs/1202.3639). The problem this solves seems very fundamental, and the result improves previous constant factor approximations, instead solving the problem optimally.
- **Strongly universal string hashing is fast**, Owen Kaser and Daniel Lemire, [arXiv:1202.4961](https://arxiv.org/abs/1202.4961). From the practical side of algorithmics, a method for hashing strings that uses less than one instruction per byte and nevertheless has guaranteed theoretical properties.
- **Known algorithms for EDGE CLIQUE COVER are probably optimal**, Marek Cygan, Marcin Pilipczuk, and Michał Pilipczuk, [arXiv:1203.1754](https://arxiv.org/abs/1203.1754) and SODA 2013. There is a simple FPT algorithm for covering the edges of a graph with a minimum number of cliques, which is however doubly exponential in the number of cliques. Apparently (see e.g. [this stackexchange question](https://stackoverflow.com/questions/12031754)) it's an old open question to do better than this. Last year it was shown that no better kernelization is possible, and now this preprint shows that double exponential is optimal regardless of whether it's achieved using a kernelization, assuming

the [exponential time hypothesis](#) is true. Apparently it's the first known (if conditional) double exponential lower bound for FPT algorithms.

- **Approximability of the vertex cover problem in power law graphs**, Mikael Gast and Mathias Hauptmann, [arXiv:1204.0982](#). I think there should be more work on algorithms that take advantage of the properties of web graphs and social networks to compute properties of these graphs more quickly than they can be computed in the worst case, and this is an example: it gets a better approximation ratio for vertex cover than can be achieved in the worst case, for any graph whose degree distribution obeys a power law.
- **Clustering is difficult only when it does not matter**, Amit Daniely, Nati Linial, and Michael Saks, [arXiv:1205.4891](#). Like the previous paper, this represents a move from worst-case complexity towards something more instance-based. The main idea here is that the only hard instances for clustering problems (under traditional worst-case algorithms) are ones in which the input is not actually clustered very well. Their definition of a "good clustering" seems very sensitive to outliers or noisy data, but perhaps that can be a subject for future work.
- **A randomized parallel algorithm with run time  $O(n^2)$  for solving an  $n \times n$  system of linear equations**, Joerg Fliege, [arXiv:1209.3995](#). The real breakthrough here is Raghavendra's layered generate-and-test algorithm for linear equations over finite fields, [described earlier this year on Richard Lipton's blog](#). This note shows how to do the same thing over the real numbers.
- **Faster deterministic fully-dynamic graph connectivity**, Christian Wulff-Nilsen, [arXiv:1209.5608](#) and SODA 2013. Ok, it's [log shaving](#) (actually loglog shaving) but on an important problem.
- **Eight-fifth approximation for TSP paths**, András Sebő, [arXiv:1209.3523](#) and IPCO 2013. I'd taught Christofides algorithm enough times to know to be careful about paths vs cycles in it, but somehow it had flown under the radar for me that the approximation ratio for paths was significantly larger.
- **Approximating  $k$ -median via pseudo-approximation**, Shi Li and Ola Svensson, [arXiv:1211.0243](#). Here the  $k$ -median problem is that of finding exactly  $k$  cluster centers minimizing the sum of distances of each point to its center, an approximation again finds exactly  $k$  centers but only approximately minimizes the sum, and a pseudo-approximation finds a number of centers that is only approximately  $k$ . As the authors say, using even one extra center may make a big difference in the objective function, so it is a bit of a surprise that pseudo-approximation can be useful in approximation for this problem.
- **Accelerating kd-tree searches for all  $k$ -nearest neighbours**. Bruce Merry, James Gain and Patrick Marais . <http://pubs.cs.uct.ac.za/archive/00000847/>

- **Approximate k-nearest neighbour based spatial clustering using k-d tree.**  
Mohammed Otair <http://arxiv.org/abs/1303.1951>