Algorithmic Applications of the Frobenius Normal Form

Piotr Sankowski   Karol Węgrzycki

Institute of Informatics
University of Warsaw

bit.ly/frobenius-form

Abstract in STACS 2017

What exactly is the Frobenius Normal Form?

$$A = UFU^{-1} = U \begin{bmatrix} C_1 & & & & 0 \\ & C_2 & & & \\ & & C_3 & & \\ & & & \ddots & \\ 0 & & & & C_k \end{bmatrix} U^{-1} \quad \text{where} \quad C_i = \begin{bmatrix} 0 & \cdots & & 0 & -c_0 \\ 1 & 0 & & 0 & -c_1 \\ & 1 & \ddots & \vdots & -c_2 \\ & & \ddots & 0 & \vdots \\ & & & 1\ \ 0 & -c_{r-2} \\ 0 & & & 1 & -c_{r-1} \end{bmatrix} \in K^{r \times r}$$

We will use the property, that we can power companion matrix in a smart way, by shifting columns.

Example:

$$C = \begin{bmatrix} 0&0&0&0&1\\1&0&0&0&2\\0&1&0&0&3\\0&0&1&0&4\\0&0&0&1&5 \end{bmatrix}, C^2 = \begin{bmatrix} 0&0&0&1&5\\0&0&0&2&11\\1&0&0&3&17\\0&1&0&4&23\\0&0&1&5&29 \end{bmatrix}, C^3 = \begin{bmatrix} 0&0&1&5&29\\0&0&2&11&63\\0&0&3&17&98\\1&0&4&23&133\\0&1&5&29&168 \end{bmatrix},$$

$$C^4 = \begin{bmatrix} 0&1&5&29&168\\0&2&11&63&365\\0&3&17&98&567\\0&4&23&13&770\\1&5&29&16&973 \end{bmatrix}, C^5 = \begin{bmatrix} 1&5&29&168&973\\2&11&63&365&2114\\3&17&98&567&3284\\4&23&133&770&4459\\5&29&168&973&5635 \end{bmatrix}$$

All Nodes Shortest Cycles

For every vertex in the graph return the length of the shortest cycle that contains it.

Date	Authors	Time	Comments
–	Naive	$O(n^3)$	–
2011	Yuster	$\tilde{O}(n^{(\omega+3)/2})$	Undirected
Nov 2017	Agarwal and Ramachandran	$\tilde{O}(n^\omega)$	Undirected
Nov 2017	**This Paper**	$\tilde{O}(n^\omega)$	**Directed**

Sets on Cycles

Determine the set of vertices $S(t)$ that lie on some cycle of length at most t.

Date	Authors	Time	Comments
2011	Yuster	$\tilde{O}(tn^\omega)$	Returns: $S(t)$
2015	Cygan et al.	$\tilde{O}(n^\omega)$	Returns: $S(t)$
2017	**This Paper**	$\tilde{O}(n^\omega)$	Returns: $S(1), \ldots, S(D)$

All Pairs All Walks

Return an array A, such that for every pair of vertices $u, v \in G$ and every $k \in \{1, \ldots, D\}$ an element $A[u, v, k]$ is the **number of distinct walks** from u to v of length k.

Naive: $O(Dn^\omega) = O(n^{3.373})$

This Paper: $O(n^3 \log n)$

Distance Queries

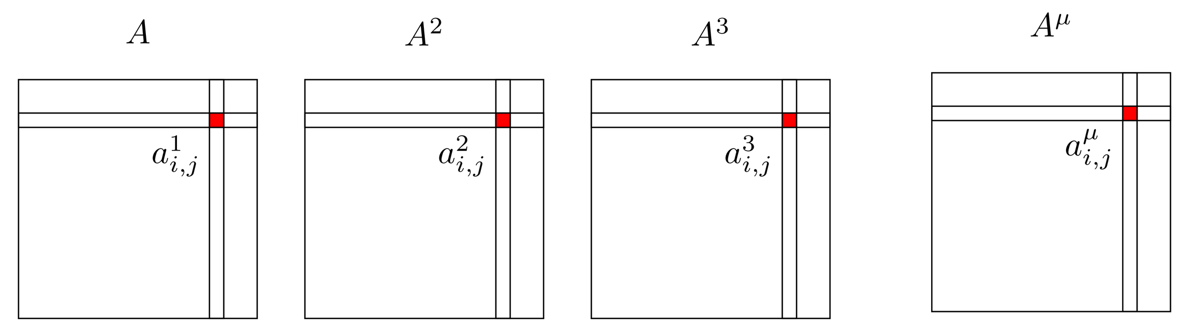
Preprocess graph in such a way, that you can answer queries about distance $\delta(u, v)$ fast.

Authors	Preprocessing	Query	Comments
Naive APSP	$O(n^{2.52})$	$O(1)$	
Yuster Zwick	$O(n^{2.38})$	$O(n)$	$\delta(u, v)$
This Paper	$\tilde{O}(n^{2.38})$	$O(n \log n)$	$\delta^1(u, v), \ldots \delta^D(u, v)$

Main Theorem

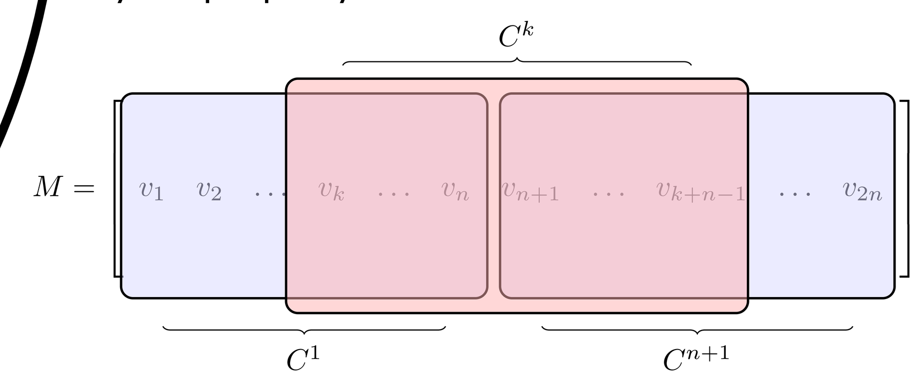
Say $a^m_{i,j}$ is an element on i-th row and j-th column of matrix A^m.
We show an algorithm that preprocess matrix A.
Then we can ask queries of form (i, j) for an elements of A.
That outputs: $a^1_{i,j}, a^2_{i,j}, \ldots, a^\mu_{i,j}$

- We need $\tilde{O}(n^\omega)$ preprocessing time,
- And $O(n \log n)$ query time.

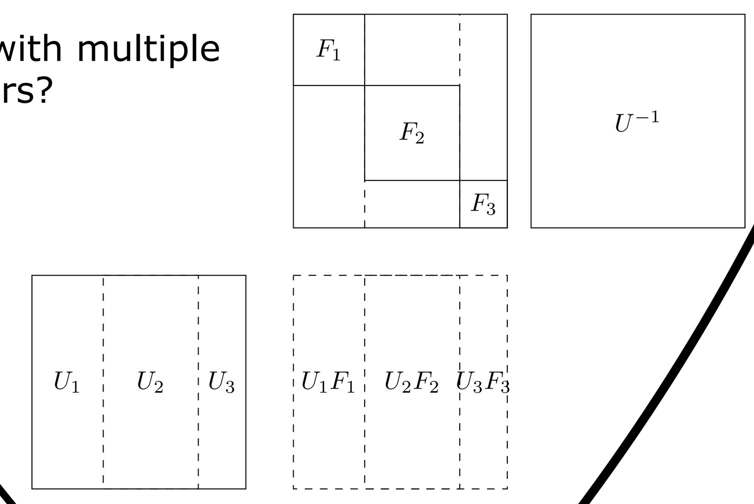


How to Prove It?

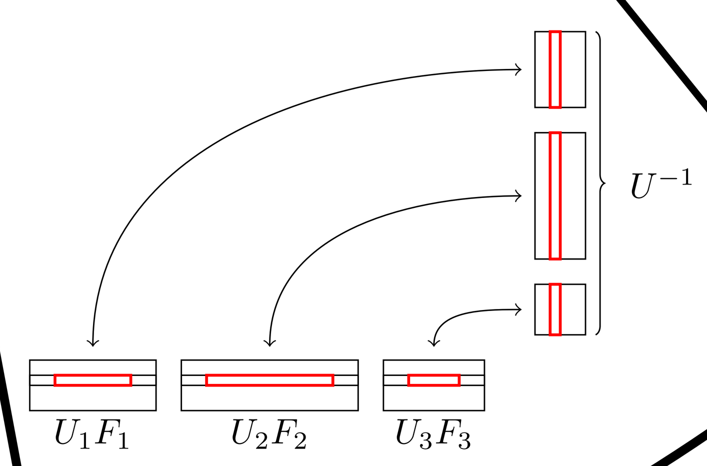
Cyclic property



How to get it with multiple invariant factors?



Using FFT allows us to answer queries in near linear time



Future Work

Dynamic APSP

In SODA 2017 an worst case $O(n^{2.5})$ update time dynamic APSP was shown. Can we parametrize it, e.g., $O(kn^2)$ for a graph parameter k? Moreover it is a major open problem to show $O(n^{2.49})$ worst case time algorithm.

Random Walker on Graph

$$P_{i,j} = \begin{cases} \frac{1}{\text{degout}(i)} & \text{if } i \to j \\ 0 & \text{otherwise} \end{cases}$$

And if we take the power of the probability $P_{i,j}$ fast then we can answer querie concerning random walker

Why should I read this paper?

Here we introduce completly new algebraic method in graph algorithms. We use simple black-box tools to show algorithms running in fast matrix multiplication time.

Moreover this paper considerably simplifies distance queries algorithms in graphs. We hope that this paper will inspire progress on All-Pairs Shortest Paths in directed, unweighted graphs.

European Research Council
Established by the European Commission

This work is part of a project TOTAL that has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No 677651).