学习资料

# Basics:

1. 本科算法：Introduction to Algorithm (<https://edutechlearners.com/download/Introduction_to_algorithms-3rd%20Edition.pdf> ),

Algorithm Design (<https://ict.iitk.ac.in/wp-content/uploads/CS345-Algorithms-II-Algorithm-Design-by-Jon-Kleinberg-Eva-Tardos.pdf>),

1. 高级算法：Randomized Algorithms (<https://rajsain.files.wordpress.com/2013/11/randomized-algorithms-motwani-and-raghavan.pdf> ), Probability and Computing (<https://www.cs.purdue.edu/homes/spa/courses/uj16/mu-book.pdf> ), Approximation Algorithms (<https://www.ics.uci.edu/~vazirani/book.pdf> )
2. 计算复杂性：Computational Complexity: A Modern Approach (<https://theory.cs.princeton.edu/complexity/book.pdf> )
3. 概率论：The probabilistic method (Alon and Spencer <http://math.bme.hu/~gabor/oktatas/SztoM/AlonSpencer.ProbMethod3ed.pdf> )
4. 谱图论：Spectral graph theory （<http://cs-www.cs.yale.edu/homes/spielman/sagt/sagt.pdf>）
5. 数据科学，机器学习理论：Foundation of Data Science ([Online version](https://www.cs.cornell.edu/jeh/book.pdf)), Mathematic foundation for data analysis ([Online version](https://mathfordata.github.io/))
6. 理论计算机科学工具包： <https://www.bilibili.com/video/BV1hy4y127kP/?spm_id_from=333.788.recommend_more_video.4>
7. differential privacy：[Differential Privacy - Resources](https://differentialprivacy.org/resources/)

## 有用的经验：

1. 如何做理论研究：<https://www.bilibili.com/video/BV1hy4y127kP/?spm_id_from=333.788.recommend_more_video.4>
2. 值得反反复复读的经验：<https://www.cs.cmu.edu/~mblum/research/pdf/grad.html>
3. 如何读文章：

3.1 <https://www.eecs.harvard.edu/~michaelm/postscripts/ReadPaper.pdf>

3.2 <https://web.stanford.edu/class/ee384m/Handouts/HowtoReadPaper.pdf>

4. 各种经验：<https://cstheory.stackexchange.com/questions/tagged/soft-question+advice-request+research-practice>

# 研究方向：

# Computational resources (Time, Space, Communication etc.) constrained -- Algorithms:

## Sublinear time algorithms:

0. 资源汇总（里面有很多资源包括课程等）：<https://sublinear.info/index.php?title=Main_Page>

0. Introduction to property testing (<https://www.wisdom.weizmann.ac.il/~oded/PDF/pt-v3.pdf> )

### On graphs:

1. Edge Sampling and Graph Parameter Estimation via Vertex Neighborhood Accesses (<https://arxiv.org/pdf/2107.03821.pdf> )
2. Massively Parallel Computation and Sublinear-Time Algorithms for Embedded Planar Graphs (<https://arxiv.org/pdf/2204.09035.pdf>)
3. Testability and local certification of monotone properties in minor-closed classes (<https://arxiv.org/pdf/2202.00543.pdf>)
4. Sublinear Time Spectral Density Estimation (<https://arxiv.org/pdf/2104.03461.pdf>)
5. Approximating the Spectrum of a Graph (<https://arxiv.org/pdf/1712.01725.pdf>)
6. Sublinear TSP (<https://arxiv.org/pdf/2203.14798.pdf> )
7. Tolerant Bipartiteness Testing in Dense Graphs (<https://arxiv.org/pdf/2204.12397.pdf> )
8. Computing and Testing Small Connectivity in Near-Linear Time and Queries via Fast Local Cut Algorithms (<https://arxiv.org/pdf/1910.14344.pdf> )
9. Finding forbidden minors in sublinear time I (<https://arxiv.org/pdf/1805.08187.pdf> also check the journal version )
10. Sublinear Algorithms for Hierarchical Clustering<https://arxiv.org/pdf/2206.07633.pdf>

### On functions:

1. VC Dimension and Distribution-Free Sample-Based Testing (https://arxiv.org/pdf/2012.03923.pdf)

## Streaming and sketching algorithms:

0. A survey: <https://people.cs.umass.edu/~mcgregor/papers/13-graphsurvey.pdf>

1. A lecture notes: <https://www.sketchingbigdata.org/fall20/lec/notes.pdf>
2. [Sketching as a Tool for Numerical Linear Algebra](http://dx.doi.org/10.1561/0400000060) (<http://www.cs.cmu.edu/afs/cs/user/dwoodruf/www/wNow3.pdf> )
3. Deterministic (1+*ε*)-Approximate Maximum Matching with 𝗉𝗈𝗅𝗒(1/*ε*)  Passes in the Semi-Streaming Model and Beyond (<https://arxiv.org/pdf/2106.04179.pdf> )
4. Discrepancy, Coresets, and Sketches in Machine Learning (<http://proceedings.mlr.press/v99/karnin19a/karnin19a.pdf> )

## Dynamic algorithms:

0. A survey: <https://researchweek.comp.nus.edu.sg/slides2021/Monika%20Henzinger.pdf>

1. A workshop: <https://sites.google.com/view/stoc22-dynamic-workshop/>
2. Dynamic Algorithms Against an Adaptive Adversary: Generic Constructions and Lower Bounds (<https://arxiv.org/pdf/2111.03980.pdf> new dynamic algorithms against adaptive adversary by using differential privacy that protect randomness)
3. Maintaining Expander Decompositions via Sparse Cuts (<https://arxiv.org/pdf/2204.02519.pdf> )

## Massively parallel computation (MPC) algorithms:

0. 讲义 <http://people.csail.mit.edu/ghaffari/MPA19/Notes/MPA.pdf>

1. Walking Randomly, Massively, and Efficiently (<https://arxiv.org/pdf/1907.05391.pdf>)
2. Efficient and Local Parallel Random Walks (<https://arxiv.org/pdf/2112.00655.pdf> )
3. Communication and Memory Efficient Testing of Discrete Distributions (<http://proceedings.mlr.press/v99/diakonikolas19a/diakonikolas19a.pdf> )
4. Almost 3-Approximate Correlation Clustering in Constant Rounds <https://arxiv.org/pdf/2205.03710.pdf>

## Adversary robust streaming algorithms

1. A workshop: <https://rajeshjayaram.com/stoc-2021-robust-streaming-workshop.html>
2. A Framework for Adversarially Robust Streaming Algorithms (<https://dl.acm.org/doi/pdf/10.1145/3498334> )
3. Tight Bounds for Adversarially Robust Streams and Sliding Windows via Difference Estimators (<https://arxiv.org/pdf/2011.07471.pdf> )
4. Adversarially Robust Streaming Algorithms via Differential Privacy (<https://arxiv.org/pdf/2004.05975.pdf> new robust streaming algorithms by using differential privacy that protect randomness)
5. Adversarial Robustness of Streaming Algorithms through Importance Sampling (<https://arxiv.org/pdf/2106.14952.pdf> )

## Time-Space tradeoff

1. Fast Learning Requires Good Memory: A Time-Space Lower Bound for Parity Learning (<https://dl.acm.org/doi/pdf/10.1145/3186563> )
2. A Time-Space Lower Bound for a Large Class of Learning Problems (<https://par.nsf.gov/servlets/purl/10064470> )
3. Memory-sample tradeoffs for linear regression with small error (<https://arxiv.org/pdf/1904.08544.pdf> )
4. Estimating Entropy of Distributions in Constant Space (<https://arxiv.org/pdf/1911.07976.pdf> )

============================

# 2. Applications constrained (Privacy, Robustness, Learning) - Algorithms

## Learning augmented algorithms:

1. A talk: [Learning-Augmented Algorithms: A New Direction?](https://www.eecs.harvard.edu/~michaelm/TALKS/ALGO2019.pptx)
2. A workshop: <http://www.mit.edu/~vakilian/ttic-workshop.html>
3. Lectures on Learning-Augmented Algorithms (<https://antoniosantoniadis.github.io/learning-augmented-algorithms.html> )
4. Learning Combinatorial Optimization Algorithms over Graphs（<https://arxiv.org/abs/1704.01665>）
5. Putting the “Learning" into Learning-Augmented Algorithms for Frequency Estimation (<http://proceedings.mlr.press/v139/du21d/du21d.pdf> )
6. The Primal-Dual method for Learning Augmented Algorithms (<https://arxiv.org/pdf/2010.11632.pdf> )

## Robust estimation:

1. Robustness Meets Algorithms (<https://dl.acm.org/doi/pdf/10.1145/3453935> The survey on high dimensional robust mean estimation)
2. Recent Advances in Algorithmic High-Dimensional Robust Statistics (<https://arxiv.org/pdf/1911.05911.pdf> )
3. Robust Estimation for Random Graphs (<https://arxiv.org/pdf/2111.05320.pdf>)

## Differential privacy:

0. A course: <http://www.gautamkamath.com/CS860-fa2020.html>

1. Differential Privacy from Locally Adjustable Graph Algorithms: k-Core Decomposition, Low Outdegree Ordering, and Densest Subgraphs
2. Differentially Private Analysis of Graphs

<https://cs-people.bu.edu/sofya/pubs/GraphPrivacyEncyclopedia.pdf>

1. Consistent Spectral Clustering of Network Block Models under Local Differential Privacy

<https://journalprivacyconfidentiality.org/index.php/jpc/article/view/811/738>

1. A book: https://www.cis.upenn.edu/~aaroth/Papers/privacybook.pdf

<https://archive.siam.org/meetings/sdm14/hardt-nikolov.pdf>

1. On the Robustness of CountSketch to Adaptive Inputs (<https://arxiv.org/pdf/2202.13736.pdf> )
2. Privately Estimating Graph Parameters in Sublinear time (<https://arxiv.org/pdf/2202.05776.pdf> )
3. Differentially Private All-Pairs Shortest Path Distances （<https://arxiv.org/pdf/2203.16476.pdf> ）
4. Locally Private k-Means Clustering (<https://jmlr.org/papers/volume22/20-721/20-721.pdf> )
5. Differentially Private Release of Synthetic Graphs (<https://theory.epfl.ch/kapralov/papers/DP_cut_apx.pdf>)
6. Differentially Private Clustering in High-Dimensional Euclidean Spaces (<https://www.cis.upenn.edu/~tbd/docs/ICML2017PrivateClustering.pdf> )
7. [Practical Differentially Private Clustering](http://ai.googleblog.com/2021/10/practical-differentially-private.html)( <https://ai.googleblog.com/2021/10/practical-differentially-private.html> )
8. Private Coresets (<http://people.csail.mit.edu/dannyf/private.pdf> )

## Clustering, dimension reduction, coresets

1. A survey: <https://link.springer.com/content/pdf/10.1007/s13218-017-0519-3.pdf>
2. The Johnson-Lindenstrauss Lemma for Clustering and Subspace Approximation: From Coresets to Dimension Reduction (<https://numbda.cs.tsinghua.edu.cn/~yuwj/TH-CPL.pdf> )
3. A New Coreset Framework for Clustering (<https://arxiv.org/pdf/2104.06133.pdf>)
4. A cost function for similarity-based hierarchical clustering (<https://arxiv.org/pdf/1510.05043.pdf> )

## Faulty oracle and active learning

1. Approximation Algorithms for Large Scale Data Analysis (A tutorial; <https://dl.acm.org/doi/pdf/10.1145/3452021.3458813> )
2. How to Design Robust Algorithms using Noisy Comparison Oracle (<https://arxiv.org/pdf/2105.05782.pdf> )

# 3. Classical graph algorithms:

1. A talk: <https://www.bilibili.com/video/BV1tZ4y1Q7qn/>
2. Maximum Flow and Minimum-Cost Flow in Almost-Linear Time (This is a recent breakthrough:<https://arxiv.org/pdf/2203.00671.pdf> )
3. Faster Maxflow via Improved Dynamic Spectral Vertex Sparsifiers (<https://arxiv.org/pdf/2112.00722.pdf> )
4. Bipartite Matching in Nearly-linear Time on Moderately Dense Graphs (<https://arxiv.org/pdf/2009.01802.pdf>)