L^3

Team Notebook

06/11/2018

Contents

1. Data Structures 1
1.1. Union Find 1
1.2. Segment Tree 1
1.3. Fenwick Tree 1
1.4. Heap 1
1.5. Balanced Binary Search Tree 1
1.6. AVL 1
1.7. Treap 1
1.8. Sparse Table 1
2. Graphs 1
2.1. Single-Source Shortest Paths 1
2.2. All-Pairs Shortest Paths 1
2.3. Strongly Connected Components 1
2.4. Cut Points and Bridges 1
2.5. Minimum Spanning Tree 1
2.6. Topological Sorting 1
2.7. Bipartite Matching 1
2.8. Maximum Flow 1
2.9. Centroid Decomposition 1
2.10. Least Common Ancestor 1
3. Strings 1
3.1. Knuth-Morris-Pratt Algorithm 1
3.2. Trie 1
3.3. Hashing 1
4. Mathematics 1
4.1. Binomial Coefficients 1
4.2. Euclidean Algorithm 1
4.3. Primality Test 1
4.4. Sieve 1
4.5. Modular Exponentiation 1
4.6. Modular Multiplicative Inverse 1
4.7. Chinese Remainder Theorem 1
4.8. Numeric Integration (Simpson's Rule) 1
4.9. Fast Fourier Transform 1
4.10. Josephus Problem 1
4.11. Number of Integer Points Below a Line 1
5. Geometry 1
5.1. Primitives 1
5.2. Lines 1
5.3. Circles 1
5.4. Polygons 1
5.5. Convex Hull (Graham's Scan) 1
5.6. Closest Pair of Points 1
5.7. Rectilinear Minimum Spanning Tree 1
6. Other Algorithms 1
6.1. 2SAT 1
6.2. Nth Permutation 1
6.3. Floyd's Cycle-Finding 2
6.4. Longest Increasing Subsequence 2

1. Data Structures

1.1. Union Find.

1.2. Segment Tree.

1.2.1. *Recursive.*

1.2.2. *Iterative.*

1.2.3. *Lazy Propagation.*

1.2.4. *Persistent.*

1.3. Fenwick Tree.

1.3.1. *Point-updates.*

1.3.2. *Range-updates.*

1.4. Heap.

1.5. Balanced Binary Search Tree.

1.6. AVL.

1.7. Treap.

1.7.1. *Explicit.*

1.7.2. *Implicit.*

1.8. Sparse Table.

2. Graphs

2.1. Single-Source Shortest Paths.

2.1.1. *Dijkstra.*

2.1.2. *Bellman-Ford.*

2.1.3. *A*.*

2.2. All-Pairs Shortest Paths.

2.2.1. *Floyd-Washall.*

2.3. Strongly Connected Components.

2.3.1. *Kosaraju.*

2.4. Cut Points and Bridges.

2.5. Minimum Spanning Tree.

2.5.1. *Kruskal.*

2.5.2. *Prim.*

2.6. Topological Sorting.

2.7. Bipartite Matching.

2.7.1. *Alternating Paths Algorithm.*

2.7.2. *Hopcroft-Karp Algorithm.*

2.8. Maximum Flow.

2.8.1. *Edmonds-Karp.*

2.8.2. *Dinic.*

2.9. Centroid Decomposition.

2.10. Least Common Ancestor.

2.10.1. *Binary Lifting.*

2.10.2. *Tarjan's Offline Algorithm.*

3. Strings

3.1. Knuth-Morris-Pratt Algorithm.

3.2. Trie.

3.3. Hashing.

4. Mathematics

4.1. Binomial Coefficients.

4.2. Euclidean Algorithm.

4.3. Primality Test.

4.3.1. *Optimized Brute Force.*

4.3.2. *Miller-Rabin.*

4.3.3. *Pollard's Rho Algorithm.*

4.4. Sieve.

4.4.1. *Sieve of Eratosthenes.*

4.4.2. *Divisor Sieve (Modified Sieve of Eratosthenes).*

4.4.3. *Phi Sieve.*

4.5. Modular Exponentiation.

4.6. Modular Multiplicative Inverse.

4.7. Chinese Remainder Theorem.

4.8. Numeric Integration (Simpson's Rule).

4.9. Fast Fourier Transform.

4.10. Josephus Problem.

4.11. Number of Integer Points Below a Line.

5. Geometry

5.1. Primitives.

5.2. Lines.

5.3. Circles.

5.4. Polygons.

5.5. Convex Hull (Graham's Scan).

5.6. Closest Pair of Points.

5.7. Rectilinear Minimum Spanning Tree.

6. Other Algorithms

6.1. 2SAT.

6.2. Nth Permutation.

6.3. **Floyd's Cycle-Finding.**

6.4. **Longest Increasing Subsequence.**