

# Algorithms and Data Structures

## Data Structures and Algorithms in Java

2-satisfiability  
 \_How to solve algorithmic problem (draft)  
 Aho-Corasick algorithm  
 Aho-Corasick simple  
 Binary heap based on ArrayList  
 Binary heap with increase priority operation  
 Binary Search  
 Binary Search Tree  
 Binomial coefficients and combinations enumeration  
 Bron-Kerbosch algorithm for maximum independent set  
 Delaunay triangulation and Voronoi diagram in  $O(N \cdot \sqrt{N})$  (with demo)  
 Delaunay triangulation in  $O(N^4)$  (with demo)  
 Determinant of a matrix by Gauss and Crout algorithms in  $O(N^3)$   
 DFS: Bridges finding  
 DFS: Cut points finding  
 DFS: Eulerian cycle  
 DFS: Strongly connected components. Kosaraju's algorithm  
 DFS: Strongly connected components. Tarjan's algorithm  
 DFS: Topological sorting  
 Disjoint-set data structure  
 Doubly Linked list  
 Drawing of connected graph with Force-Based method  
 Dynamic programming: Convex Hull Optimization  
 Dynamic programming: domino fill  
 Dynamic programming: edit distance in  $O(N^2)$   
 Dynamic programming: longest common subsequence in  $O(N^2)$   
 Dynamic programming: longest increasing subsequence in  $O(N^2)$   
 Dynamic programming: number of perfect matchings  
 Dynamic programming: number of solutions of linear equality  
 Dynamic programming: optimal matrix chain multiplication in  $O(N^3)$   
 Enumeration of arrangements  
 Enumeration of partitions  
 Enumeration of permutations  
 Euclidean algorithm. GCD, LCM, modular inverse, Chinese remainder theorem  
 Expression parser: Shunting-yard algorithm  
 Factorization in  $O(\sqrt{N})$   
 Fenwick tree 2D for sum  
 Fenwick tree for sum  
 Fenwick tree for sum with extended operations  
 Gaussian elimination algorithm in  $O(N^3)$   
 Geometry: Circle  
 Geometry: Class Complex  
 Geometry: Line  
 Gradient Descent  
 Graph  
 Greedy graph coloring in  $O(E \cdot \log V)$   
 Hashing on strings  
 Heavy-light tree decomposition for edges or vertices  
 Hungarian algorithm for assignment problem  
 Kd-tree for nearest neighbour query in  $O(\log N)$  on average  
 Kd-tree for rectangular query in  $O(\sqrt{N})$   
 Kth order statistic in  $O(N)$  on average  
 LCA. Lowest common ancestor on tree in  $O(\log N)$   
 LCA: Sparse Table  
 LinkCut tree - dynamic tree connectivity  
 LinkCut tree - dynamic tree lca  
 Longest increasing subsequence in  $O(N \cdot \log N)$   
 Matrices  
 Max Rectangle  
 Maximum flow of minimum cost in  $O(V^3 \cdot \text{FLOW})$   
 Maximum flow of minimum cost with Bellman-Ford in  $O(\min(E^2 \cdot V^2, E \cdot V \cdot \text{FLOW}))$   
 Maximum flow of minimum cost with potentials in  $O(\min(E^2 \cdot V \cdot \log V, E \cdot \log V \cdot \text{FLOW}))$   
 Maximum flow. Dinic's algorithm in  $O(V^2 \cdot E)$   
 Maximum flow. Edmonds-Karp algorithm in  $O(\min(E^2 \cdot V, E \cdot \text{FLOW}))$   
 Maximum flow. Ford-Fulkerson algorithm in  $O(V^2 \cdot \text{FLOW})$   
 Maximum flow. Push-relabel algorithm in  $O(V^3)$   
 Maximum matching for bipartite graph. Kuhn's algorithm in  $O(E \cdot V)$   
 Maximum matching for bipartite graph. Kuhn's algorithm in  $O(V^3)$   
 Maximum matching for general graph. Edmond's algorithm in  $O(V^3)$   
 Maximum matching for general graph. Randomized algorithm in  $O(V^3)$   
 Meet in the middle  
 Mergeable heap. A heap with merge, add, removeMin operation in  $O(\log N)$   
 Minimum spanning tree. Prim's algorithm in  $O(E \cdot \log V)$   
 Minimum spanning tree. Prim's algorithm in  $O(V^2)$   
 Pair (std::pair analog)  
 Persistent Tree  
 Prefix tree (Trie)

## Data Structures and Algorithms in Java >

### Segment Tree 2D without recursion with single addition for maximum

```

public class SegmentTree2DFastAddMax {
    int n;
    int m;
    int[][] t;

    public SegmentTree2DFastAddMax(int n, int m) {
        this.n = n;
        this.m = m;
        t = new int[2 * n][2 * m];
    }

    public int max(int x1, int y1, int x2, int y2) {
        int res = Integer.MIN_VALUE;
        for (x1 += n, x2 += n; x1 <= x2; x1 = (x1 + 1) >> 1, x2 = (x2 - 1) >> 1, i2 = (i2 + 1) >> 1, i2 = (i2 - 1) >> 1) {
            for (int i1 = y1 + m, i2 = y2 + m; i1 <= i2; i1 = (i1 + 1) >> 1, i2 = (i2 - 1) >> 1) {
                res = Math.max(res, t[x1][i1]);
                res = Math.max(res, t[x1][i2]);
                res = Math.max(res, t[x2][i1]);
                res = Math.max(res, t[x2][i2]);
            }
        }
        return res;
    }

    public void add(int x, int y, int value) {
        t[x + n][y + m] += value;
        for (x += n; x > 0; x >= 1) {
            if (x > 1)
                t[x >> 1][y + m] = Math.max(t[x][y + m], t[x >> 1][y + m]);
            for (int i = y + m; i > 1; i >= 1) {
                t[x][i >> 1] = Math.max(t[x][i], t[x][i >> 1]);
            }
        }
    }

    public int get(int x, int y) {
        return t[x + n][y + m];
    }

    public void set(int x, int y, int value) {
        add(x, y, -get(x, y) + value);
    }

    // Usage example
    public static void main(String[] args) {
        SegmentTree2DFastAddMax t = new SegmentTree2DFastAddMax(2, 1);
        t.add(0, 0, 1);
        System.out.println(1 == t.max(0, 0, 1, 0));
    }
}

```

## Comments

<div>Prime numbers, sieve of Eratosthenes, Euler's totient function</div> <div>Quadtree for rectangular queries in <math>O(\min(n, N+M))</math></div> <div>Queue with minimum query in <math>O(1)</math></div> <div>Random permutations and arrangements</div> <div>Random tree and graph generation. Prüfer code</div> <div>Rational numbers class</div> <div>RMQ: Sparse Table</div> <div>Searching substrings in <math>O(N)</math>. Knuth–Morris–Pratt algorithm + prefix function</div> <div><b>Segment Tree 2D without recursion with single addition for maximum</b></div> <div>Segment Tree with interval modification</div> <div>Segment Tree with interval modification without recursion</div> <div>Segment Tree. Simple implementation</div> <div>Shortest Hamiltonian cycle (TSP) in <math>O(2^N \cdot N^2)</math></div> <div>Shortest Hamiltonian path in <math>O(2^N \cdot N^2)</math></div> <div>Shortest paths. Bellman–Ford algorithm in <math>O(V^3)</math>. Negative cycle detection.</div> <div>Shortest paths. Dijkstra's algorithm in <math>O(E \cdot \log V)</math></div> <div>Shortest paths. Floyd–Warshall algorithm in <math>O(V^3)</math></div> <div>Simplex algorithm</div> <div>Sorting algorithms: qsort, merge, bubble, selection, insertion, counting, radix</div> <div>Suffix Array in <math>O(N \cdot \log N)</math> and LCP in <math>O(N)</math></div> <div>Suffix Array in <math>O(N \cdot \log N^2)</math></div> <div>Suffix automaton</div> <div>Suffix tree. Ukkonen's algorithm in <math>O(N \cdot \text{alphabetSize})</math></div> <div>Travelling salesman problem: genetic algorithm (with demo)</div> <div>Travelling salesman problem: simulated annealing (with demo)</div> <div>Treap as a set with kth-element operation</div> <div>Treap with implicit key with interval modification</div> <div>Tree Centers</div> <div>Universal sqrt-decomposition for queries</div>	
<div><b>Classic problems</b></div> <div>Longest palindromic subsequence</div>	
<div><b>Data Structures and Algorithms in C++</b></div> <div>Arbitrary-precision arithmetic</div> <div>Binary exponentiation algorithm</div> <div>C++ comparators</div> <div>Class Scanner for fast input</div> <div>Diameter of a planar point set in <math>O(N \cdot \log N)</math> with rotating calipers method</div> <div>Disjoint-set data structure with rank heuristic</div> <div>Fenwick tree for sum</div> <div>Fenwick tree for sum on Map</div> <div>Geometry convex hull: Graham-Andrew algorithm in <math>O(N \cdot \log N)</math></div> <div>Geometry: finding a pair of intersected segments in <math>O(N \cdot \log N)</math></div> <div>Kd-tree for nearest neighbour query in <math>O(\log N)</math> on average</div> <div>Laguerre's method of polynomial roots finding</div> <div>Matrices</div> <div>Maximum flow of minimum cost in <math>O(\min(E^2 \cdot V \log V, E \cdot \log V \cdot \text{FLOW}))</math></div> <div>Maximum flow. Dinic's algorithm in <math>O(V^2 \cdot E)</math></div> <div>Maximum matching for bipartite graph. Hopcroft-Karp algorithm in <math>O(E \cdot \sqrt{V})</math></div> <div>Minimum spanning tree. Prim's algorithm in <math>O(E \cdot \log V)</math></div> <div>Segment Tree with interval modification</div> <div>Shortest paths. Dijkstra's algorithm with binary heap in <math>O(E \cdot \log V)</math></div> <div>Shortest paths. Dijkstra's algorithm with priority_queue or set in <math>O(E \cdot \log V)</math></div> <div>Sieve of Eratosthenes in <math>O(N \cdot \log \log N)</math></div> <div>SSE Instructions</div> <div>Suffix Array and LCP in <math>O(N)</math>. Algorithm DC3</div> <div>Treap with implicit key with interval modification</div> <div>Tree isomorphism</div>	
<div><b>Sitemap</b></div>	
<div><b>Recent site activity</b></div> <div><a href="#">Data Structures and Algorithms in Java</a> edited by Andrey Naumenko</div> <div><a href="#">How to solve algorithmic problem (draft)</a> edited by Andrey Naumenko</div> <div><a href="#">Data Structures and Algorithms in Java</a> edited by Andrey Naumenko</div> <div><a href="#">Segment Tree with interval modification without recursion</a> edited by Andrey Naumenko</div> <div><a href="#">Segment Tree with interval modification</a> edited by Andrey Naumenko</div> <div><a href="#">View All</a></div>	