

# Contents

<b>Basic</b>	<b>3</b>
vimrc . . . . .	3
Debug Macro . . . . .	3
Increase Stack . . . . .	3
Pragma Optimization . . . . .	3
IO Optimization . . . . .	4
SVG Writer . . . . .	4
<b>Data Structure</b>	<b>4</b>
Dark Magic . . . . .	4
Link-Cut Tree . . . . .	4
LiChao Segment Tree . . . . .	4
Treap . . . . .	5
Linear Basis . . . . .	5
Binary Search On Segtree . . . . .	5
<b>Matching &amp; Flow</b>	<b>5</b>
HopcroftKarp . . . . .	5
Kuhn Munkres . . . . .	5
Flow Models . . . . .	6
Dinic . . . . .	6
HLPP . . . . .	6
Global Min-Cut . . . . .	6
GomoryHu Tree . . . . .	6
Minimum Cost Max Flow . . . . .	7
Dijkstra Cost Flow . . . . .	7
Minimum Cost Circulation . . . . .	7
Capacity Scaling . . . . .	7
General Graph Matching . . . . .	7
Weighted Matching . . . . .	7
<b>Graph</b>	<b>8</b>
2-SAT (SCC) . . . . .	8
BCC . . . . .	8
Round Square Tree . . . . .	8
Edge TCC . . . . .	8
DMST . . . . .	8
Dominator Tree . . . . .	8
Edge Coloring . . . . .	9
Centroid Decomposition . . . . .	9
Lowbit Decomposition . . . . .	9
Virtual Tree . . . . .	9
Tree Hashing . . . . .	9
Mo's Algorithm on Tree . . . . .	9
Count Cycles . . . . .	10
MaximalClique . . . . .	10
Maximum Clique (Dyn) . . . . .	10
Minimum Mean Cycle . . . . .	10
<b>Math</b>	<b>10</b>
Common bounds . . . . .	10
Stirling Number . . . . .	10
Extended Euler . . . . .	11
Extended Floor Sum . . . . .	11
Integer Division . . . . .	11
Floor Sum . . . . .	11
ModMin . . . . .	11
Floor Monoid Product . . . . .	11

ax + by = gcd . . . . .	12
Chinese Remainder . . . . .	12
DiscreteLog . . . . .	12
Quadratic Residue . . . . .	12
FWT . . . . .	12
Packed FFT . . . . .	12
CRT for arbitrary mod . . . . .	13
NTT . . . . .	13
FPS . . . . .	13
Partition Number . . . . .	13
Pi Count . . . . .	13
Miller Rabin . . . . .	14
Pollard Rho . . . . .	14
Berlekamp Massey . . . . .	14
Charateristic Polynomial . . . . .	14
Simplex / Simplex Construction . . . . .	14
Adaptive Simpson . . . . .	14
<b>Geometry</b>	<b>15</b>
Basic Geometry . . . . .	15
2D Convex Hull . . . . .	15
2D Farthest Pair . . . . .	15
MinMax Enclosing Rect . . . . .	15
Minkowski Sum . . . . .	15
Segment Intersection . . . . .	16
Half Plane Intersection . . . . .	16
SegmentDist . . . . .	16
Rotating Sweep Line . . . . .	16
Polygon Cut . . . . .	16
Point in Simple Polygon . . . . .	16
Point in Hull (Fast) . . . . .	17
Tangent of Points To Hull . . . . .	17
Circle Class & Intersection . . . . .	17
Circle Common Tangent . . . . .	17
Line-Circle Intersection . . . . .	17
Poly-Circle Intersection . . . . .	17
Minimum Covering Circle . . . . .	18
Circle Union . . . . .	18
Polygon Union . . . . .	18
3D Point . . . . .	18
3D projection . . . . .	18
3D Convex Hull . . . . .	18
Delaunay . . . . .	19
kd Tree (Nearest Point) . . . . .	19
kd Closest Pair (3D ver.) . . . . .	19
Simulated Annealing . . . . .	19
Triangle Centers . . . . .	19
<b>Stringology</b>	<b>20</b>
Hash . . . . .	20
Description . . . . .	20
Test Status . . . . .	20
Suffix Array . . . . .	20
Ex SAM . . . . .	20
Z value . . . . .	20
Manacher . . . . .	20
Lyndon Factorization . . . . .	20
Main Lorentz . . . . .	21
BWT . . . . .	21

Palindromic Tree . . . . .	21
<b>Misc</b>	<b>21</b>
Theorems . . . . .	21
Weight Matroid Intersection . . . . .	22
Stable Marriage . . . . .	22
Bitset LCS . . . . .	22
Prefix Substring LCS . . . . .	22
Convex 1D/1D DP . . . . .	22
ConvexHull Optimization . . . . .	22
De-Bruijn . . . . .	23
Josephus Problem . . . . .	23
N Queens Problem . . . . .	23
Manhattan MST . . . . .	23
Tree Knapsack . . . . .	23
Binary Search On Fraction . . . . .	23
Barret Reduction . . . . .	24
Montgomery Multiplication . . . . .	24

## Basic

### vimrc

#### Description

vimrc.

1. Be careful of the version (currently `gnu++20` for WF)
2. `setxkbmap` command should be executed in terminal or `smt`.

#### Test Status

No test needed

### Debug Macro

#### Description

Debug code for dumping information.

#### Test Status

No test needed.

### Increase Stack

#### Description

Increase the stack size

#### Test Status

Not even used

### Pragma Optimization

#### Description

Magic Pragas. It depends to choose `Ofast` or `O3`. For target related stuff, adding `arch=skylake` should work (no need for others). Also, a [way](#) to avoid [denormal numbers](#). `0x8000` for FTZ and `0x0040` for DAZ. [Intel Compiler Docs](#). Only works for SSE/AVX stuff.

### **Test Status**

Rarely used, no test

## **I/O Optimization**

### **Description**

I/O bounded program needs this sweet optimization.

### **Test Status**

Rarely used, no test.

## **SVG Writer**

### **Description**

A helper to generate SVG. Support Line, Circle, and Text. Should adjust sizes properly.

### **Test Status**

No Test

## **Data Structure**

### **Dark Magic**

### **Description**

PBDS classes/functions. ordered set and mergable heap are the useful ones.

### **Test Status**

No test.

### **Link-Cut Tree**

### **Description**

$O(Q \log N)$  operations on path query. Supports link or cut edge.  
Subtree queries are tricky.

### **Test Status**

CF 603E. Passed [dynamic\\_tree\\_vertex\\_set\\_path\\_composite](#) and [dynamic\\_tree\\_vertex\\_add\\_subtree\\_sum](#).

### **LiChao Segment Tree**

### **Description**

Maintain the upper envelope of lines.  
TODO: is extended version needed?

### **Test Status**

Used in some contest.

## Treap

### Description

treap. For persistent, should not use `pri`.

### Test Status

Rarely used. Need test?

## Linear Basis

### Description

Given a set of integers: - `query_kth` to find the  $k$ -th integer in the (sorted) set of XOR combination of the integers with  $v$ . - The `second` field is for range XOR basis query or smt, greedily maintained in `insert` function.

### Test Status

- [ABC223 H](#)
- kth problem [1st Hunger Games S](#)
- maybe need a combined problem?

## Binary Search On Segtree

### Description

Binary search on ZKW segtree. `sz` should be power of 2 (be careful of other parts!).

### Test Status

Passed [Quick Sort](#)

## Matching & Flow

### HopcroftKarp

#### Description

An  $O(|E|\sqrt{|V|})$  bipartite matching algorithm. Basically a low constant Dinic's algorithm.

Number of matching saved in `ans`, and the corresponding matching saved in `l` and `r`. ~~Not sure about what `a` and `p` does.~~ `a` and `p` are auxiliary array when doing BFS.

### Test Status

Tested on [Library Checker](#)

### Kuhn Munkres

#### Description

KM algo.

### Test Status

Passed [UOJ 80](#) and [Library Checker](#).

## Flow Models

### Description

Some models. Need check.

### Test Status

TODO

## Dinic

### Description

Dinic with capacity scaling. See [this](#) and [this](#).  $O(VE \log U)$  and  $\Theta(\text{acceptable})$  in practice.

### Test Status

Passed [luogu P3376](#).

## HLPP

### Description

HLPP algo with gap heuristics.

Theoretical complexity is  $O(V^2\sqrt{E})$ . But heuristic is powerful!

Note: Lowest Label Push Relabel is  $O(\sqrt{VE})$  on bipartite matching graph.

### Test Status

[LOJ 127](#) and [library checker bipartite matching](#). Passed [Matching on Bipartite Graph](#).

## Global Min-Cut

### Description

Stoer-Wagner algorithm solves the minimum cut problem in undirected weighted graphs with non-negative weights. Our code looks like an  $O(N^3)$  implementation.

### Test Status

Passed [luogu](#) Didn't find a  $O(VE + V^2 \log V)$  version.

## GomoryHu Tree

### Description

For a given non-negative weighted tree, this algorithm returns a weighted tree (Gomory-Hu Tree). For any  $s, t$ , the minimum  $s$ - $t$  cut in the original graph is equal to the minimum values among the path between  $s$  and  $t$  in the Gomory-Hu Tree.

Runs in  $(|V| - 1) \times O(\text{maflow})$ .

Need to adapt current Dinic's algorithm.

Something I don't understand: In the Gomory-Hu tree, for any pair of vertices not just the size of the minimum cut between them is equal to the size of the minimum cut in the original graph (as Wikipedia claims), but also the minimum cut itself (as a partition of the vertex set into two). ([Petr's blog](#))

Fun Fact: Gomory-Hu Tree can be computed in almost linear time. (see [this](#))

### Test Status

Passed [CF 343E](#).

## Minimum Cost Max Flow

### Description

Successive Shortest Path Algorithm using SPFA (Bellman-Ford algorithm).

### Test Status

Passed [LibreOJ 102](#). Testdata in LOJ is not strong in general.

## Dijkstra Cost Flow

### Description

Successive Shortest Path Algorithm using Dijkstra's algorithm.

### Test Status

Tested on [ARC122 F](#) and [LibreOJ 102](#)

## Minimum Cost Circulation

### Description

Network simplex method. Exponential time complexity, but it runs not too slow in practice.

### Test Status

Tested on [UOJ #487](#), [UOJ #680](#), and [LibreOJ 102](#). [min\\_cost\\_b\\_flow](#). **Cannot** pass [QOJ 7185](#)

## Capacity Scaling

### Description

### Test Status

[min\\_cost\\_b\\_flow](#)

## General Graph Matching

### Description

Matching in  $O(|V|^3)$ . [ref-slide](#)

### Test Status

Tested on [Library Checker](#).

## Weighted Matching

### Description

Weighted matching in  $O(|V|^3)$ . [ref-slide](#)

### Test Status

Tested on [Library Checker](#)

## Graph

### 2-SAT (SCC)

#### Description

Kosaraju and 2-SAT construction. Don't forget we can do bitset optimization. To use 2-SAT,  $2i$  and  $2i+1$  represents  $x$  and  $\neg x$ .  $x \vee x$  or  $\neg x \vee \neg x$  is OK.

#### Test Status

Passed CSES Giant Pizza and [CF Radio Stations](#). Passed [library checker](#).

### BCC

#### Description

Gives AP and bridge and `bcc_id`. `bcc_id[edge_id]` is the bcc of the edge.

#### Test Status

Passed [Two-Edge-Connected-Components](#) and [Biconnected Components](#). `is_ap` function is not tested.

### Round Square Tree

#### Description

Or block-cut-tree. Useful tree for "simple path" queries. There will be at most  $2N$  vertices in the new tree.

#### Test Status

Passed [2020 Shanghai K](#) Passed [Biconnected Components](#)

### Edge TCC

#### Description

Edge triconnected component.

#### Test Status

Passed [yosupo library checker](#).

### DMST

#### Description

Directed Minimum Spanning Tree in  $O(E \log^2 E)$ . Use mergable heap instead of small-to-big for better complexity?

#### Test Status

Passed [yosupo library checker](#) [CF 100307 D](#)

### Dominator Tree

#### Description

Dominator tree in  $O(E \log V)$ . The ancestor relation on the tree is the "must-pass-from-source" relation in original graph.



### Test Status

Passed [yosupo library checker](#).

## Edge Coloring

### Description

[Misra & Gries edge coloring algorithm](#). Runs in  $O(NM)$

### Test Status

Passed [NCPC 2018 G](#).

## Centroid Decomposition

### Description

Mark a vertex or query the sum of distance from a vertex to all marked vertices.

### Test Status

Need rewrite or smt.

## Lowbit Decomposition

### Description

Some chain decomposition of tree.

### Test Status

TODO

## Virtual Tree

### Description

Dependency: `lca`. Gives the critical nodes of given subset. Always include the original root. The edges are given in rooted tree format.

### Test Status

Used in contest. TODO.

## Tree Hashing

### Description

Some PRNG random hash.

### Test Status

Passed [UOJ 763](#) and [library checker](#).

## Mo's Algorithm on Tree

### Description

Pseudo code of mo's algo on tree. `push` means XOR the contribution.

## Test Status

TODO

## Count Cycles

### Description

Count 3-cycle and 4-cycle in  $O(M\sqrt{M})$ .

## Test Status

Passed [CCPC Guangzhou](#).

## MaximalClique

### Description

Enumerate maximal clique. Time complexity  $O(n3^{n/3})$  or  $O(nC)$  where  $C$  is the number of such cliques.

## Test Status

Can run on  $n = 80$  on [TIOJ](#).

## Maximum Clique (Dyn)

### Description

Get maximum clique with ?? time complexity.

## Test Status

kactl says it can run on  $n = 155$ . For  $n = 100$  on P0J, runs in 32ms. Passed [library checker](#).

## Minimum Mean Cycle

### Description

$O(V(V + E))$  find min mean cycle. Too rare to use so needs shorten.

## Test Status

Passed a UVa problem with  $n = 50$ .

## Math

### Common bounds

### Description

Partition function, divisor function and catalan number.

## Test Status

No test.

## Stirling Number

### Description

Stirling number formula. Do we need this?

**Test Status**

No test.

**Extended Euler****Description**

A formula.

**Test Status**

No test.

**Extended Floor Sum****Description**

A recursion formula.

**Test Status**

No test.

**Integer Division****Description**

C++ integer division to normal integer division.

**Test Status**

Copied from 8BQube

**Floor Sum****Description**

Calculate  $\sum_{i=0}^{n-1} \lfloor \frac{ai+b}{m} \rfloor$ .

**Test Status**

Passed [yosupo judge](#) (negative coefficient not tested).

**ModMin****Description**

Return the minimum  $x \geq 0$  such that  $l \leq ax \bmod m \leq r$ .

**Test Status**

Tested on [SEERC'20 G](#)

**Floor Monoid Product****Description**

萬能歐幾里得 [ref1](#) [ref2](#)

### Test Status

<https://judge.yosupo.jp/submission/185615>  
<https://loj.ac/s/1986411>

<https://www.luogu.com.cn/record/144016921>

### **ax + by = gcd**

#### Description

exgcd algorithm.

### Test Status

See CRT section.

### Chinese Remainder

#### Description

Solves  $x \equiv r_1 \pmod{m_1}$  and  $x \equiv r_2 \pmod{m_2}$ . If no solution, returns false

### Test Status

Passed luogu P4777.

### DiscreteLog

#### Description

BSGS algorithm.

### Test Status

Passed [yosupo judge](#)

### Quadratic Residue

#### Description

Square root under modulo prime.

### Test Status

Passed [yosupo judge](#)

### FWT

#### Description

Bitwise XOR/AND/OR convolution.

### Test Status

Passed yosupo judge, [XOR](#) and [AND](#) version.

### Packed FFT

#### Description

Make FFT precision better. reference: - [淺談 FFT](#) - [題解 P4245](#)

## Test Status

Passed [convolution mod](#) with long double. For  $N = 524288$ , - normal NTT (998244353): ~230ms - three-mod-NTT: ~430ms - [convolution\\_mod](#) ~1000ms with long double (AC), 400ms with double (WA) - [convolution](#): ~800ms with long double (WA)

## CRT for arbitrary mod

### Description

CRT for three-mod-NTT.

### Test Status

Passed [yosupo judge](#).

## NTT

### Description

NTT. Can be modified to FFT easily.

### Test Status

Passed yosupo judge. See also "CRT for arbitrary mod".

## FPS

### Description

Common Formal Power Series operations. Exp and Pow are relatively slow at yosupo library checker.

Do we need [Consecutive Terms of Linear Recurrent Sequence?](#)

### Test Status

[Inv](#) [Ln](#) [Exp](#) [Pow](#) [Sqrt](#) [Eval](#) [DivMod](#) [LinearRecursionKth](#)

## Partition Number

### Description

Calculate first  $N$  partition number in  $O(N\sqrt{N})$ .

### Test Status

Passed [yosupo judge](#)  $N = 500000$  in 557ms.

## Pi Count

### Description

Count prime in sublinear time. The code is copied from 8BQube and simplified.

### Test Status

Passed [yosupo judge](#)

## Miller Rabin

### Description

Prime detect. Be careful about mpow and mmul.

### Test Status

Passed [yosupo judge](#) in 1632ms ( $10^5$  tests). w/ Montgomery Multiplication runs in [230ms](#).

## Pollard Rho

### Description

Factorization. Be careful about mpow and mmul.

### Test Status

Passed [yosupo judge](#) in 313ms (100 tests). w/ Montgomery Multiplication runs in [73ms](#)

## Berlekamp Massey

### Description

BM algo.

### Test Status

Passed [yosupo judge](#).

## Charateristic Polynomial

### Description

Calculate the charateristic polynomial of matrix in  $O(N^3)$ .

### Test Status

Passed 2021 PTZ Korea and yosupo library checker.

## Simplex / Simplex Construction

### Description

Linear programming.

### Test Status

TODO.

## Adaptive Simpson

### Description

Simpson integration method. Unknown time complexity.

### Test Status

Passed [Two Cylinders](#)

# Geometry

## Basic Geometry

### Description

- `sgn` `cross` `dot` `ori`
- `quad` `argCmp` all-integer angle compare.
- `area` be careful of type.
- `rot90` multiply by  $i$  (or left turn 90 degree)
- `project` projection onto a vector

### Test Status

No test. Used extensively in other template. TODO Center of polygon needs test.

## 2D Convex Hull

### Description

Returns strict convex hull of given points. The result is counter-clockwise and the first point is the lex-min point. Be carefule about edge case (0/1/2/3 points on CV)

### Test Status

Used in some contest.

## 2D Farthest Pair

### Description

Rotating caliper algorithm. Requires the input hull be strictly convex.

### Test Status

Passed A0J CGL.

## MinMax Enclosing Rect

### Description

Rotating caliper, but with more pointers.

### Test Status

Passed UVA 819

## Minkowski Sum

### Description

Minkowski sum of two convex hulls.

### Test Status

Used in some contest. TODO.

## Segment Intersection

### Description

Check whether the segment intersects. Touching at the ends counts. Be careful about edge case like parallel, does touching at ends count, ... Can be modified to `Ray` class or `Line` class.

To get the intersection point, check next part (HPI)

### Test Status

Used in many contest. Passed A0J CGL.

## Half Plane Intersection

### Description

Calculate the area of half-plane-intersection. The result lines will be in `q` (this is why we need the reference). Result lines maybe wrong if the intersection area doesn't have positive area.

### Test Status

Passed 2020 Nordic NCPC Big brother. Used in many contest.

## SegmentDist

### Description

Distance from point to segment and segment to segment. Can be used in checking sausage intersection.

### Test Status

Passed Q0J 2444 and PTZ 19 summer D3.

## Rotating Sweep Line

### Description

A skeleton of rotating sweep line. Support colinear cases.

### Test Status

Passed [NAIPC 2016 G](#)

## Polygon Cut

### Description

Cut simple polygon by a line.

### Test Status

Copied from kactl. TODO.

## Point in Simple Polygon

### Description

Testing PIP.



### **Test Status**

Used in some contest. TODO.

## **Point in Hull (Fast)**

### **Description**

Testing PIH in  $O(\log N)$ .

### **Test Status**

[Enclosure](#) Used in some contest.

## **Tangent of Points To Hull**

### **Description**

Tangent of point to hull in  $O(\log N)$ . Requires the hull to be strictly convex. Can be modified to find extreme point on hull.

### **Test Status**

[Enclosure](#)

## **Circle Class & Intersection**

### **Description**

Definition of `Cir` and some intersection function.

### **Test Status**

Passed A0J CGL.

## **Circle Common Tangent**

### **Description**

Common tangent point of circle.

### **Test Status**

Passed A0J CGL and [CF 128E](#).

## **Line-Circle Intersection**

### **Description**

The point of intersection of line and circle.

### **Test Status**

TODO.

## **Poly-Circle Intersection**

### **Description**

The intersection area of a circle and a simple polygon.

### Test Status

Passed A0J CGL\_7\_H. Copied from 8BQube and they say it passed HDU2892.

## Minimum Covering Circle

### Description

Get minimum covering circle in  $O(N)$  expected time. Also gives the circumcenter formula.

### Test Status

Passed TI0J 1093, luogu P1742

## Circle Union

### Description

Calculate the area that covered by at least  $k$  circle for each  $k$ . Time complexity  $O(N^2 \log N)$ .

### Test Status

Passed SP0J.

## Polygon Union

### Description

Union area of simple polygon.

### Test Status

TODO.

## 3D Point

### Description

Basic 3d point. - cross - triple product - rotate around an axis

### Test Status

`rotate_around` is copied from NaCl. Others are tested by 3d hull.

## 3D projection

### Description

Get the 2d coordinate of the projection of a point  $p$  onto plane  $q^T x = 0$ .

### Test Status

Passed [stars in a can](#).

## 3D Convex Hull

### Description

Return the face of 3d convex hull of  $N$  points. There will be  $O(N)$  faces and time complexity is  $O(N^2)$ . Be careful of degenerate cases.

### **Test Status**

Passed SP0J and [stars in a can](#). Passed [HDU 3662](#). (need to combine coplanar triangles to one face).

## **Delaunay**

### **Description**

Delaunay triangulation.

Usage TODO.

### **Test Status**

Passed [Brazil subregional](#).

## **kd Tree (Nearest Point)**

### **Description**

KD Tree nearest point query.

### **Test Status**

TODO

## **kd Closest Pair (3D ver.)**

### **Description**

3d closest pair

### **Test Status**

Correct, but might be too slow. Can pass [TIOJ](#) using fast hash table.  
Need more test.

## **Simulated Annealing**

### **Description**

A skeleton of simulated annealing

### **Test Status**

TODO.

## **Triangle Centers**

### **Description**

Triangle centers formula.

### **Test Status**

No test.

# Stringology

## Hash

### Description

Rolling-hash algorithm

### Test Status

Used in contest. No test.

## Suffix Array

### Description

SA-IS algorithm. Complexity:  $O(N + C)$

### Test Status

Tested on [Suffix Array](#) and [Number of Substrings](#).

## Ex SAM

### Description

Don't know how to use.

### Test Status

Copied from 8bq

## Z value

### Description

Z algorithm

### Test Status

Tested on [Library Checker](#)

## Manacher

### Description

Find maximal palindrome for each index.

### Test Status

Tested on [Library Checker](#)

## Lyndon Factorization

### Description

A string is called simple (or a Lyndon word), if it is strictly smaller than any of its own nontrivial suffixes. The Lyndon factorization of the string  $s$  is a factorization  $s = w_1 w_2 \dots w_k$ , where all strings  $w_i$  are simple, and they are in non-increasing order  $w_1 \geq w_2 \geq \dots \geq w_k$ .

Duval algorithm:  $O(N)$ .

## Test Status

Tested @ luogu 6114, 1368 & UVA 719. Passed [Library Checker](#)

## Main Lorentz

### Description

A repetition is two occurrences of a string in a row. The challenge is to find all repetitions in a given string  $s$ .

The algorithm described here was published in 1982 by Main and Lorentz.

Time complexity:  $O(N \log N)$

Every  $[l, r]$  in  $\text{rep}[i]$  satisfies that if  $p \in [l, r]$  then  $s[p, p+i) = s[p+i, p+2i)$ .

## Test Status

Passed [CF 104508J](#). This problem is prepared with this code, but some SA solutions also passes.

## BWT

### Description

Burrows-Wheeler transform is done by sorting all the circular shifts of a text in lexicographic order and by extracting the last column and the index of the original string in the set of sorted permutations of  $S$ .

Good for run-length encoding?

## Test Status

Passed UVa 632 and UVa 741

## Palindromic Tree

### Description

Check [OI Wiki](#)

Don't know how to use.

## Test Status

TODO

## Misc

### Theorems

### Description

Theorems.

## Test Status

No test.

## Weight Matroid Intersection

### Description

Almost an implementation.

### Test Status

Copied from NaCl

## Stable Marriage

### Description

Stable Marriage algo.

### Test Status

No test needed.

## Bitset LCS

### Description

$O(n^2/w)$ . need hand-written bitset (needs subtraction) TODO: Find a way to recover the answer. [Prob](#)

### Test Status

Passed [LibreOJ #6564](#)

## Prefix Substring LCS

### Description

Calculate the LCS of a prefix of  $S$  and a substring of  $T$  in  $O((|S||T| + Q) \log |T|)$

### Test Status

Passed [yosupo library checker](#). Copied from 8BQube.

## Convex 1D/1D DP

### Description

1D/1D optimization.

### Test Status

[TIOJ 烏龜疊疊樂](#)

## ConvexHull Optimization

### Description

Maintain upper envelope of lines.

### Test Status

Passed [yosupo library checker](#).

## De-Bruijn

### Description

De-Bruijn sequence construction

### Test Status

Passed CSES, [regional prob](#) and local test.

## Josephus Problem

### Description

Josephus problem  $O(N)$  and faster algo ( $O(M \log N)$ ).

### Test Status

Passed [2018 Asia Nanjing](#).

## N Queens Problem

### Description

N Queens Problem construction

### Test Status

Not even used or tested.

## Manhattan MST

### Description

Minimum Spanning Tree of manhattan distance.

### Test Status

Passed [yosupo library checker](#).

## Tree Knapsack

### Description

TODO don't know its usage

### Test Status

Not even used or tested.

## Binary Search On Fraction

### Description

Binary search on stern-borcot tree, binary search over  $p/q$  such that  $0 \leq p, q \leq N$ .

### Test Status

TODO. Copied from NaCl.

## **Barret Reduction**

### **Description**

Fast modulo operation of non-constexpr constant. Only able to handle int-size modulo.

### **Test Status**

Copied from kactl. Guess it's ok to have no test.

## **Montgomery Multiplication**

### **Description**

Fast modulo operation of non-constexpr constant. Only able to handle odd modulo.

### **Test Status**

Tested with MillerRabin and PollardRho.