

世界の天才プログラマに挑戦！

『2位じゃダメなんですか？』

プログラミングコンテストで、

世界のトップコーダーに挑んだ、

アルゴリズムの数々を紹介します。

Challenge the world's genius programmers!

"Isn't it good for second place?"

Introducing a number of algorithms

that challenged the world's top coders

in programming contests.

「プログラミングでやる格闘技」 - Martial arts programming

- **プログラミング・コンテスト**

課題を解くコードを書き、時間内に多く解いた者が勝ち。

→ Google Code Jam、TopCoder、ACM/ICPC など

- **オンライン・ジャッジシステム**

課題を解くコードを書き、サーバにUPすると、

実行スピードに基づいて順位が決定する。

→ Sphere Online Judge (SPOJ) など

- **The result of the contest is 2/370 in Japan and 71/82k in the world.**

Binary Search

クエリで指定されたソート済みで**ダブリ**のあるデータ(10^5)の位置を答える。(標準関数呼び出す前にダブリを除く工夫が必要です)

Answer the location of the sorted, **doubled** data specified in the query. (It is necessary to devise to eliminate duplication before calling the standard function)

チュートリアル課題ですが、タイムアウトが異様に多いため、トラップが、仕掛けられていると見て、3階層インデックスで解きました。This is a tutorial task, but because there are many timeouts, I saw that the trap was set, and solved it with a 3-level index.

Binary search statistics & best solutions						
Users accepted	Submissions	Accepted	Wrong Answer	Compile Error	Runtime Error	Time Limit Exceeded
487	5221	1126	1978	444	453	1195
All ADA DOC ASM AWK BASH BF C C# C++ 4.3.2 C++ 4.0.0-8 C99 strict CLPS CLOJ LISP sbcl LISP clisp D ERL F# FORT GO HASK ICON ICK JAR JAVA JS LUA NEM NICE NODEJS CAML PAS fpc PAS gpc PDF PERL PERL 6 PHP PIKE PS PRLG PYTH 2.7 PYTH 3.2.3 PYTH 3.2.3 n RUBY SCALA SCM guile SCM qobi SED ST TCL TECS TEXT WSPC						
RANK	DATE	USER	RESULT	TIME	MEM	LANG
1	2014-07-05 06:20:10	kunio himel	accepted edit run	0.41	2.5M	C
2	2014-01-17 13:03:33	Syed Sarfaraz Akhtar	accepted	0.46	6.4M	C++ 4.3.2
3	2012-03-15 19:58:14	setsquare	accepted	0.53	2.0M	C

3階層インデックス作成部分のコードです。エレガントですね。

This is the code for the 3-level index creation part. It's elegant.

```
void index(register int key) {
    INDEX1[index1] = key;
    if (0 == (index1++ & INDEX_BIT_MASK)) {
        INDEX2[index2] = key;
        if (0 == (index2++ & INDEX_BIT_MASK))
            INDEX3[index3++] = key;
    }
}
```

Binary Search のスピードが気になったのでコードを書いてみました。
 結果は、2位で、タイムアウトが多いのは、前処理の誤りのようです。
 I was worried about the speed of Binary Search, so I wrote a
 code. The result is the second place, and it seems that a lot of
 timeouts are due to an error in preprocessing.

```
int binarySearch(register int key) {
  register int low = 0;
  register int high = element - 1;
  while (low <= high) {
    register int mid = (low + high) / 2;
    register int midVal = ELEMENTS[mid];
    if (midVal < key)
      low = mid + 1;
    else if (midVal > key)
      high = mid - 1;
    else
      return LOCATION[mid]; // key found
  }
  return -1; // key not found.
}
```

Binary search (/problems/BSEARCH1) statistics & best solutions

Users accepted	Submissions	Accepted	Wrong Answer	Compile Error	Runtime Error	Time Limit Exceeded
2843	30754	6514	10749	2586	2039	8818

RANK	DATE	USER	RESULT	TIME	MEM	LANG
1	2013-05-29 14:55:38	kunio himei (/users/himei/)	accepted	0.04	2.8M	C
2	2018-11-14 07:54:42	kunio himei (/users/himei/)	accepted	0.05	17M	C++
3	2018-11-14 08:10:37	kunio himei (/users/himei/)	accepted	0.06	3.5M	C++ 4.3.2
4	2013-10-21 02:44:41	./ (/users/mernet/)	accepted	0.07	4.7M	C++ 4.3.2

$$X_{n+1} = X_n - (X_n^2 - 2) / 2 X_n$$

- 1.41421356... (Up to 2 million digits)
- Source code size limit: 4096 bytes

乗算の高速化アルゴリズム (Faster multiplication)

- ①筆算方式 $O(n^2)$
- ②Karatsuba方式 (通常4回必要な乗算を3回で計算 $O(n^{1.585})$)
- ③FFT - 高速フーリエ変換 $O(n \cdot \log(n))$

RANK	DATE	USER	RESULT	TIME	MEM	LANG
1	2014-05-13 08:34:54	kunio himei	520359 edit run	13.78	177M	JAVA
2	2013-09-20 14:47:19	C. Guerber	90064	14.43	178M	JAVA
3	2011-12-07 22:06:26	welfremd	66666	18.66	178M	JAVA
RANK	DATE	USER	RESULT	TIME	MEM	LANG
1	2007-04-29 07:04:27	Hideyuki Tanaka	2000000	16.89	48M	C
2	2010-06-05 07:58:45	Mainak Biswas	2000000	11.14	48M	C
3	2008-10-08 08:29:10	Paulo Roberto Santos de Sousa	1599990	17.42	26M	C
4	2013-11-04 05:34:51	kunio himei	674538 edit run	15.49	39M	C

C言語のトップは、東大の数値計算の専門家です。ここまでのどりに着くのに、2年かかりました。

もう1ステップ上まで行きたかったのですが、アルゴリズムの本に載っていた、FFTのチューニングでは、どうにもなりません。そこで、宗旨替えすることにしました。

The top of the C language is the University of Tokyo's numerical calculation specialist.

It took 2 years to get here.

I wanted to go one step further, but the tuning of FFT, which was mentioned in the book on algorithms, did not help.

So I decided to change my religion.

Pike



RANK	DATE	USER	RESULT	TIME	MEM	LANG
1	2012-12-01 20:17:22	Robert Gerbicz	2000000	2.67	8.2M	PIKE
2	2015-01-16 07:47:49	kunio himei	2000000	1.72	9.9M	PIKE
3	2009-09-28 18:10:16	numerix	1820002	2.75	10M	PIKE
4	2011-09-11 18:27:43	Michael Kharitonov	1820002	2.74	8.9M	PIKE
5	2011-12-13 14:17:17	XeRoNIX	1818002	2.75	8.9M	PIKE
6	2014-02-26 02:42:49	Mitch Schwartz	1802242	2.72	8.9M	PIKE
7	2013-08-14 15:46:03	John Doe	1800002	2.75	8.2M	PIKE
8	2013-03-19 14:05:28	[Retired] Blue.Mary	1777002	2.72	8.2M	PIKE
9	2011-05-21 10:59:09	Vikhyat Korrapati	1710002	2.75	8.8M	PIKE
10	2011-09-29 01:23:13	Meritz Gronbach	1600002	2.34	7.5M	PIKE
11	2013-04-07 14:00:13	(Tjandra Satria Gunawan) (曾殺昆)	1150000	2.51	7.6M	PIKE
12	2013-04-22 16:01:10	NIKA	1048577	2.65	7.6M	PIKE
13	2014-11-22 16:52:30	Francky	200685	1.20	7.6M	PIKE

予想通り成功しました。(コードは、21Stepです)

ところで、3位以下のトップコーダ達は、何をしているのでしょうか？ その秘密は。コードよく見ると、

小数点以下を整数演算していることです。

(これは、Calcで、シミュレーションを行っている時に気づきました)

Successful as expected. (The code is 21Steps)

By the way, what are the top 3 coders below?

What is the secret? If you look closely at the code,

It is an integer operation below the decimal point.

(This was noticed when doing a simulation in Calc)



// PIKE

```
Gmp.mpz P, Q, XX, X = Gmp.mpz(
/*3908*/"xwf8wu07adrr252ycf5lf1rkojip848tumv2cct34790csm ... ", 36);
int main() {
    int n = X->size(10),
        i = 2;
    P = Gmp.mpz(pow(10, n-1));
    while (10 >= i++) {
        n <<= 1;
        Q = P->pow(2) * 10;
        XX = X->pow(2);           // XX = X ^ 2
        XX /= pow(10, XX->size(10) - n);
        X *= (15 * Q) - XX;      // X = 1.5 - 0.499...
        X /= P;
        P = Q;
    }
    string R = (string) (X << 1);           // sqrt(2) = 2 * X
    R[0] = '.';
    write("1%s", R);
    return 0;
}
```

先頭の呪文は、36進数で書いた初期値(4k)です。

Pikeは、GNU Multi-Precision Library (GMP)を内臓しています。

The first spell is the default value (4k) in 36 hex.

Pike includes the GNU Multi-Precision Library (GMP).

Factorial challenge (階乗)

与えられた桁数に相当する $n!$ の n を求める。

Find n of $n!$ Corresponding to the given number of digits.

$(1 \leq x \leq 10^9)$

- スターリングの公式 (Stirling's formula)

$$n! \approx \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \left(1 + \frac{1}{12n} + \frac{1}{288n^2} - \frac{139}{51840n^3}\right)$$

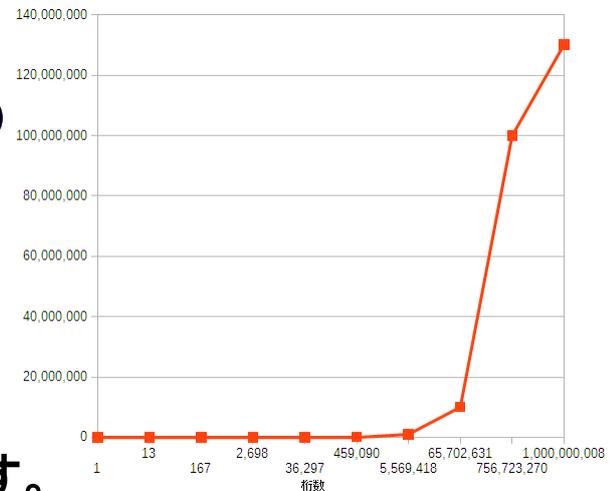
与えられる、桁数の範囲が大きいため、
キャッシュが使えず、難易度の高い課題です。

Given the large range of digits,

This is a very difficult problem because cash cannot be used.

リラククスしている時にアルゴリズムが閃きました。禪で言う悟りです。

The algorithm flashed while I was relaxing. Enlightenment in Zen.



	RANK	DATE	USER	RESULT	TIME	MEM	LANG
Java	1	2013-05-18 03:18:21	kunio himei	accepted edit run	0.55	177M	JAVA
	2	2011-03-15 21:42:18	Ефиме	accepted	1.80	178M	JAVA
	3	2012-06-19 23:57:34	Darko Aleksic	accepted	2.17	177M	JAVA
	RANK	DATE	USER	RESULT	TIME	MEM	LANG
	261	2010-10-17 20:12:44	numerix	accepted	8.02	9.0M	RUBY
	262	2012-07-23 03:05:36	prateek	accepted	8.12	2.6M	C++ 4.3.2
	263	2011-01-30 11:49:57	Tarun Chabarwal	accepted	8.40	1.6M	C
	264	2012-10-22 07:09:41	Masao Hara	accepted	10.65	178M	JAVA
	265	2011-05-12 21:36:58	The Bartender	accepted	13.05	3.6M	HASK
	266	2012-10-14 17:25:36	shrey	accepted	13.68	178M	JAVA
	267	2012-10-14 17:29:42	Nihav Jain	accepted	14.33	178M	JAVA
	268	2012-06-26 17:50:51	Geoff Scrace	accepted	14.39	42M	PYTH 2.7
AWK	269	2013-05-04 03:39:25	kunio himei	accepted	14.72	2.1M	AWK

苦勞した課題が何と、"AWK"でブービーです。上にいるのは、
ライバルになった、ドイツ人で、Pysonマスターと呼ばれています。

What a difficult task is "AWK" is booby.

Above is A rival German, called Pyson Master.

```

# AWK
BEGIN {
    split("1 13 167 2698 36297 459090 5569418 65702631 756723270 1000000008",
    FACT)
    split("1 15 104 1043 10159 100503 1000618 10006510 100019089 130202809",
    FACTn)
    NFACT_2PI = 2 * 3.141592653589793
    NFACT_LN10 = log(10)
    NFACT_1DLN10 = 1 / NFACT_LN10
}
1==NR { T = $1; next
}
{
    print lookup($1)
    if (NR > T) exit 0
}
# Stirling's formula
function nfact(n, xx, ii) {
    xx = (n * log(n)*NFACT_1DLN10) + (0.5 * log(NFACT_2PI * n) *
    NFACT_1DLN10) + ((-n + (1.0 / (12.0 * n))) / NFACT_LN10)
    ii = int(xx)
    return (0 == (xx - ii)) ? ii : (ii + 1)
}
function lookup(key, low, lowFact, mid, midFact, f, dd, rate) {
    if (1 == key) return 3
    if (3 == key) return 6
    for (f = 10; key < FACT[f]; --f) ;
    low = FACT[f]; lowFact = FACTn[f]
    rate = (FACTn[f + 1] - lowFact) / (FACT[f + 1] - low)
    midFact = lowFact + int(rate * (key - low))
    while (1) {
        mid = nfact(midFact)
        if (key == mid) return midFact
        dd = int((key - mid) * rate)
        if ((1 >= dd) && (-1 <= dd)) {
            if (key < mid) {
                do mid = nfact(--midFact); while (key < mid)
                return midFact
            }
            do mid = nfact(++midFact); while (key > mid)
            return (key == mid) ? midFact : (midFact - 1)
        }
        midFact += dd
    }
}
}

```

C言語で、2位に付けました。トップとの違いは、スパイスの違いか、プラットフォームの違いです (メモリサイズに注目)
プラットフォームの違いなら、宗旨替えする必要があります。

In C language 2nd place. The difference from the top is the difference of the spice or the platform (note the memory size) If the platform is different, I need to change my religion.

RANK	DATE	USER	RESULT	TIME	MEM	LANG
1	2010-11-28 20:00:43	numerix	accepted	0.26	240k	PAS fpc
2	2013-05-18 03:43:44	kunio himei	accepted edit run	0.28	1.7M	C
3	2011-07-17 13:30:01	Ashutosh Kumar	accepted	0.31	2.8M	C++ 4.3.2
4	2012-07-31 21:04:58	Krystian Plackowski	accepted	0.40	1.6M	C
5	2010-11-29 19:28:57	Michael Kharitonov	accepted	0.43	2.8M	C++ 4.3.2
6	2012-04-08 19:03:54	Krystian Plackowski	accepted	0.43	2.6M	C++ 4.3.2
7	2013-05-20 10:03:48	kunio himei	accepted edit run	0.52	177M	JAVA

真数の積は、対数の和に変換される。逆に(底が同じ)対数の和は、真数の積に変換される

$$\log_a(xy) = \log_a x + \log_a y$$

Users accepted	Submissions	Accepted	Wrong Answer	Compile Error	Runtime Error	Time Limit Exceeded
254	1700	556	370	127	191	456

All ADA DOC ASM AWK BASH BF C C# C++ 4.3.2 C++ 4.0.0-8 C99 strict CLPS CLOJ LISP sbcl LISP clisp D ERL F#
 FORT GO HASK ICON ICK JAR JAVA JS LUA NEM NICE CAML PAS fpc PAS gpc PERL PERL 6 PHP PIKE PS PRLG PYTH 2.7
 PYTH 3.2.3 PYTH 3.2.3 n RUBY SCALA SCM guile SCM qobi SED ST TCL TECS TEXT WSPC

Previous [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) Next >

RANK	DATE	USER	RESULT	TIME	MEM	LANG
1	2013-05-24 13:19:32	kunio himei	accepted edit run	0.25	1.7M	C
2	2010-11-28 20:00:43	numerix	accepted	0.26	240k	PAS fpc
3	2011-07-17 13:30:01	Ashutosh Kumar	accepted	0.31	2.8M	C++ 4.3.2
4	2012-07-31 21:04:58	Krystian Plackowski	accepted	0.40	1.6M	C
5	2010-11-29 19:28:57	Michael Kharitonov	accepted	0.43	2.8M	C++ 4.3.2
6	2012-04-08 19:03:54	Krystian Plackowski	accepted	0.43	2.6M	C++ 4.3.2
7	2013-05-20 10:03:48	kunio himei	accepted edit run	0.52	177M	JAVA

スターリングの式を対数定理を使用して変形し見事トップになりました。
The Stirling approximation was transformed using the logarithm theorem to become the top.

しかし1週間後、ドイツ人が気づき反撃してきました。
 そこからは、応酬合戦となり、結果、3ラウンド両者KOとなりました。
 しかし、こんな怪物にケツをとられたままでは、気持ちが収まらないので、トヨタ方式で、しらみつぶしに改善しました。
 最後に気づいたのは、スターリングの近似を丸のみしていることです。第4項を、削除して見事に引き離しました。

But a week later, the Germans noticed and counterattacked.
 From there, a reciprocal fight was held, and as a result, both rounds became KO.

However, if the ass is taken by such a monster, my feelings will not go away, so the Toyota method has been used to improve it.

The last thing I noticed is that the Stirling approximation is round only is. Item 4 has been removed and brilliantly separated.

もおあかん、最悪や！ (Oh No, the worst!)

RANK	DATE	USER	RESULT	TIME	MEM	LANG
1	2013-06-13 11:27:19	kunio himei	accepted edit run	0.17	1.7M	C
2	2013-06-13 17:29:47	numerix	accepted	0.17	264k	PAS fpc

枯れない、知恵の泉 (The fountain of wisdom that will not wither)

$$n! \approx \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \left(1 + \frac{1}{12n} + \frac{1}{288n^2} - \frac{139}{51840n^3}\right)$$



「来ちゃったね。」 混浴で連泊◎ 2泊以上で特典、大幅割引でお得!

RANK	DATE	USER	RESULT	TIME	MEM	LANG
1	2013-07-31 00:38:01	kunio himei	accepted edit run	0.16	1.7M	C
2	2013-06-13 17:29:47	numerix	accepted	0.17	264k	PAS fpc
3	2011-07-17 13:30:01	Ashutosh Kumar	accepted	0.31	2.8M	C++ 4.3.2
4	2012-07-31 21:04:58	Krystian Plackowski	accepted	0.40	1.6M	C

アイデアは、2~3個で、枯れ果てますが、プレッシャーを与え続けると枯れません。

The idea is a few and it will die, but if you keep applying pressure, it will not die.

Finding maximum possible number

与えられた数値(1 ~ 10^5 桁) から、K桁削除して出来る、最大の数値を答えよ。

What is the largest number that can be obtained by removing K digits from the given number (1 to 10^5 digits) ?

Users accepted	Submissions	Accepted	Wrong Answer	Compile Error	Runtime Error	Time Limit Exceeded
328	3215	747	1030	205	394	837

RANK	DATE	USER	RESULT	TIME	MEM	LANG
1	2013-02-11 08:11:29	(Tjandra Satria Gunawan) (曾殺昆)	accepted	0.06	2.7M	C++ 4.0.0-8
2	2013-05-24 10:21:41	kunio himei	accepted edit run	0.07	1.7M	C
3	2013-02-11 07:56:21	(Tjandra Satria Gunawan) (曾殺昆)	accepted	0.08	1.7M	C
4	2013-02-11 08:10:26	(Tjandra Satria Gunawan) (曾殺昆)	accepted	0.09	1.7M	C99 strict
5	2013-02-11 07:58:02	(Tjandra Satria Gunawan) (曾殺昆)	accepted	0.10	2.7M	C++ 4.3.2
6	2013-05-24 15:53:44	kunio himei	accepted edit run	0.13	2.7M	C++ 4.0.0-8
7	2013-05-02 18:37:54	abdou 00	accepted	0.19	2.6M	C++ 4.3.2

アルゴリズムを考えていると混乱する課題です。

C言語が C++(インドネシアの学生)に負けるのを初めて見ました。

情報教育は、インドネシアの方がハングリーです。

ちなみにドイツは、人口は下ですが成績は上です。

日本人は、TOP(京大)と数値計算の人(東大)を見かけただけです。

This is a confusing task when thinking about algorithms.

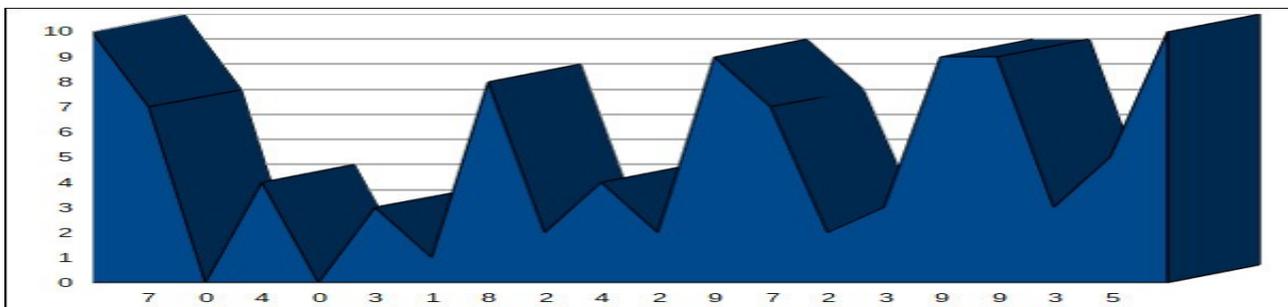
I saw C language lose to C++(Indonesian student) for the first time.

Information education is more hungry in Indonesia.

By the way, Germany has a lower population but higher grades.

Japanese people only saw TOP (Kyoto Univ.) And people of

numerical calculation (Univ. Of Tokyo).



High-speed one-pass algorithm

- ①データの前後を番兵(最大値 10)で囲む(境界判定が不要)
- ②面グラフ化し、前面と背面にアクリル板を張り、水槽を作る。
- ③先頭からスタートし、平地か下り坂なら次に進む。
- ④上り坂なら、次の地点が水没する直前まで自分自身に水を注ぎ後方の水没したデータを削除する。
- ⑤指定桁数削除するまで繰り返す。

(下り坂で何もしないのは、谷底から削除するため)

更に高速化するために、削除済み判定を不要にします。

(コンピュータの先読みキャッシュは分岐するとクリアされます。IF文を控える)

これは、下り坂で、データを後方に移動させることにより回避できます。

- ① Surround the data with sentinels (maximum value 10)
(no boundary judgment required)
- ② Make an area graph and put acrylic plates on the front and back to make an aquarium.
- ③ Start from the beginning and proceed to the next if it is flat or downhill.
- ④ If it's an uphill slope, pour water on yourself until the next point is submerged and delete the submerged data behind.
- ⑤ Repeat until the specified number of digits is deleted.

(The reason for doing nothing downhill is to remove it from the valley bottom)

To make it even faster, the deletion completion judgment is not necessary.

(The look-ahead cache of the computer is cleared when branching. Refrain from the IF statement.)

This can be avoided on the downhill by moving the data backwards.

```

void maxNum(char *begin, char *end, const int kk) {
    #define GURDE '~'
    register char *deletefirst;
    char *deletelast;
    begin[-1] = end[0] = GURDE;
    { // delete:
        int k = kk;
        register char *pi = begin;
        while (pi[0] >= pi[1]) ++pi; // 枝刈り(Pruning)
        for (deletefirst = &pi[-1]; ; ++pi) {
            register char next = pi[1];
            if (pi[0] < next) {
                deletelast = pi;
                if (0 >= --k) break;
                while (*deletefirst < next) {
                    --deletefirst;
                    if (0 >= --k) goto deletend;
                }
            } else *++deletefirst = pi[0];
        }
    } deletend: ;
    {
        register int tail = end - deletelast - 1;
        memmove(&end[-tail-kk], end - tail, (size_t) tail);
        begin[-1] = '\n';
    }
}

```

優れたアルゴリズムは、エレガントで感動を覚えます。
このアルゴリズムは、歩いているときに思いつきました。
ウォーキングは、健康と瞑想に最適です。
ちなみに、『虚無僧』がやっているのは、『歩行禅』です。

Excellent algorithms are elegant and impressive.

This algorithm came up when walking.

Walking is great for health and meditation.

By the way, what "Void priest" is doing is "Walking Zen".