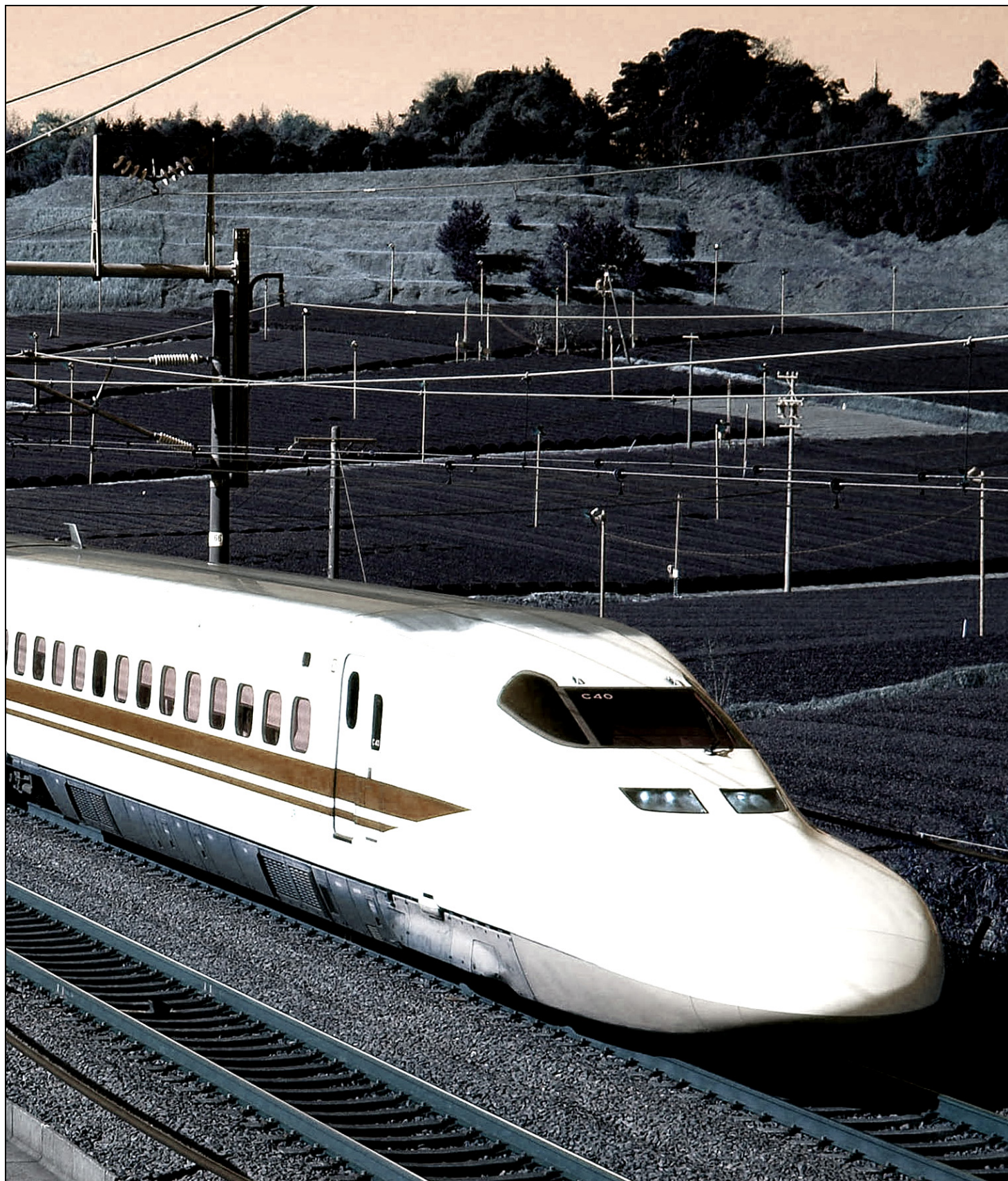


*Kazahana a.k.a. Windbloom a.k.a. Snowflake a.k.a. Snowflowerwind a.k.a. Aerialflower*



*Shinkansen Nozomi/Tsubame/Hikari/Hayabusa/Komachi/Kodama/ ... Kazahana  
Superexpress (Hope)/(Swallow)/(Light)/(Peregrine)/(a belle)/(an echo)/ ... (Windbloom)*

An x-gram superfast full-text 'on the fly' suggesteress using exact & wildcards & Levenshtein distance matching  
written in C, HEXADECAD-Threaded Linux & Windows executables

Free download at: [www.sanmayce.com/Downloads/Kazahana.zip](http://www.sanmayce.com/Downloads/Kazahana.zip)

Below, an actual snowflake (taken from Snow White's blog) under a microscope! Breathtaking and unique!



I couldn't miss that one of my oldest callnames shares a common glyph (WIND) with Kazahana, in addition it turns out that an anime character exist with the same name translated as 'windbloom', immediately I saw the connection:  
Kazahana - wind/aerial flower

One song from my childhood is still very dear to me: 'Diana Express - Severina', the world is really small: 'Diana Express' was (she is no more, she is in Trains' Heaven, Bulgaria is falling apart) the name of one of Bulgarian shinkansens (in Japanese: new rail line, in Bulgaria we simply say 'влакът стрела' i.e. 'the arrow train' not bullet train nor high velocity train, because Diana is the goddess of hunting, she is depicted as semi-naked beauty launching arrows gracefully and presumably with god-like accuracy), where 'SEVERINA' is the name of a girl (made of snow, the lyrics don't tell literally or metaphorically) coming every winter from the North, the closest equivalent of 'Severina' being 'Northina'.

Lyrics

Mitko Shterev - keyboards

Illya Angelov - lead vocal & guitar

Диана Експрес - Северина / Diana Express - Severina

Северина, момиче от сняг / Severina a girl made of snow  
всяка зима е северен знак / every winter she is a northern sign  
аз го имам в песен от юг / I have it in song from south  
Северина - радост за друг / Severina - a joy not for me

И като сняг тихо вали / And like a snow she silently comes  
вик от мойта любов / a scream from my love  
и се топи и навява тъга / and she melts and brings sadness  
песента ми за теб / my song for you

Северина, момиче от сняг / Severina a snow girl  
на приказна фея / she is fabulous fairy's  
е северен знак / northern sign  
целува ме бързо / she kisses me quickly  
и по снега тръгва зима / and winter start marching on  
бяла тъга / white sadness

I've heard that Eskimo people have more than 200 words for snow, this trumps even the sensitivity of Japanese people who are best known for their reverence to Nature, just a few snow related ones:

hatsuyuki : first snow (of season)

hyouden : field of eternal snow

koyuki : light snow

ooyuki : heavy snow

setsuzou : snow sculpture

shinshin : sound of heavy snow-fall

shinshin : mind body

yukionna : snow woman, fairy

Mutsi-mutsi, 'shinshin' is all-zen, in my view, any language lacking a word for sound of snowfalling quickly must incorporate it and fill the GAP.

In my language 'snowflake' is described with 'снежинка', whereas 'Snowwhite' with 'Снежанка', no bias here: these two Bulgarian words are fantabulous and so ringy, they are feminine (English in that respect fails to connote the most beautiful facet: tenderness), another lovely variant is the Russian 'Черыпочка', Russian is a brother language yet the 'snowflake' counterpart eludes me.

what 'Kazahana' really means? I asked this on a Japanese forum, but couldn't get the exhaustive definition, yet:

*"kazahana" (or "kazabana" - same word different reading, probably depending on the locality) is a meteorological phenomenon in which snow falling in a location quite remote from you is blown by strong winds and flutter in the sunny sky above you. Therefore, it is "snow flower," and "wind flower" and "snowing on a clear day." This doesn't happen everywhere but occurs in certain parts of Japan where the conditions are right for the phenomenon to happen. I've seen it in Kyoto and Gunma, but not in Tokyo where I live. Because of the beauty and romance of snowflakes glittering in the sun, Kazahana is popularly used as names of products, stores, etc. That is the reason why you saw it used for a doll.*

I believe there is even deeper (going back to the 'anima/soul' roots) meaning, namely 'animated-flower' i.e. 'soulful-flower'. I did my own investigation:

KAZAHANA - 風花 【かざはな】

wind - 風 can also be read "fuu" as in 台風 (taifuu / typhoon)

flower - 花

snow - 雪 【ゆき】

snowflakes/snowflowers - 雪花

**anima**, noun. E20.

[Latin = (i) air, breath, life; (ii) mind, soul.]

Psychoanalysis. The inner self (opp. persona). Also, the source of the feminine component of a personality. Cf. ANIMUS.

/SOED/

**anima**, noun & adjective. ME.

[As adjective from Old & mod. French, or Latin animalis having vital breath, (in medieval Latin) bestial, from

ANIMA: see -AL1; partly attrib. use of the noun. As noun ult. from Latin animal for animale use as noun of neut. adjective.]

/SOED/



**anima** n.  
1. The inner self of an individual; the soul.  
2. In Jungian psychology:  
a. The unconscious or true inner self of an individual, as opposed to the persona, or outer aspect of the personality.  
b. The feminine inner personality, as present in the unconscious of the male. It is in contrast to the animus, which represents masculine characteristics.  
[Latin; see anā- in Indo-European roots.]  
/Heritage/

**anə-**  
To breathe. Oldest form \*ə2enə1-, colored to \*ə2anə1-.  
Suffixed form \*anə-mo-.  
a. ANIMA, ANIMADVERT, ANIMAL, ANIMATE, ANIMATO, ANIMISM, ANIMOSITY, ANIMUS; EQUANIMITY, LONGANIMITY, MAGNANIMOUS, PUSILLANIMOUS, UNANIMOUS, from Latin animus, reason, mind, spirit, and anima, soul, spirit, life, breath;  
b. ANEMO-, ANEMONE, from Greek anemos, wind.  
[Pokorny 3. an(ə)- 38.]  
/Heritage/

Wikipedia defines '**anemone**' as:  
According to the Oxford English Dictionary, Greek anemōnē means "daughter of the wind", from ánemos "wind" + feminine patronymic suffix -ōnē.  
HERITAGE defines it as:

**anemone** n.  
1. Any of various perennial herbs of the genus Anemone, native chiefly to northern temperate regions and having palmately lobed leaves and large flowers with showy sepals. Also called windflower.  
2. A sea anemone.  
[Latin anemōnē, from Greek, probably from anemos, wind (perhaps because the petals are lost easily in wind); see anə- in Indo-European roots.]

To summarize:  
Both Latin & Greek converge:  
- 'ANIMA' (Latin: air, breath, life, mind, soul);  
- 'ANEMOS' (Greek: wind).

Literal:  
Kaza|Kaze|Fuu + Hana = wind + Flower = windflower = wind of snowflakes/snowflowers = snowflakes/snowflowers wind  
Metaphorical:  
Soul + Flower = The pure feminine inner personality = flower full of soul = soul of tender beauty = snowflower soul = snowwhite soul = pure and sparkling soul  
It rings a bell, SAKURA's petals dying still young, still full of life.  
Have you seen that sad ceremony where people release flying glowing lanterns into the sky - as if the souls of their passed friends ascending.  
Finally, snowflakes in a sunny day or fallen sakura's petals or flying lanterns, they are all souls, all true.

First, all results below are reproduceable with this test package: kazahana\_r1-++fix+nowait\_VS\_grep.7z  
On my T7500 2200MHz 4MB L2 cache processor, Windows 7 64bit, the results for next 3 patterns are:

16-threaded Exact search for '**ramjet**' into 889,537,624 bytes long file '4andabove\_Gamera.tar.2.sorted':

Allocating Master-Buffer 1536KB ... OK  
Kazahana: Dumped xgrams: 49  
Kazahana: Performance: 1,011 KB/clock  
Kazahana: Performance: Total/fread() clocks: 859/437  
Kazahana: Performance: I/O time, i.e. fread() time, is 50 percents  
Kazahana: Done.  
Timer 9.01 : Igor Pavlov : Public domain : 2009-05-31  
Kernel Time = 0.546 = 57%  
User Time = 1.138 = 119%  
Process Time = 1.684 = 176%  
Global Time = 0.953 = 100%

16-threaded Exact search for '**metal\_fatigue**' into 889,537,624 bytes long file '4andabove\_Gamera.tar.2.sorted':

Allocating Master-Buffer 1536KB ... OK  
Kazahana: Dumped xgrams: 1  
Kazahana: Performance: 1,235 KB/clock  
Kazahana: Performance: Total/fread() clocks: 703/484  
Kazahana: Performance: I/O time, i.e. fread() time, is 68 percents  
Kazahana: Done.  
Timer 9.01 : Igor Pavlov : Public domain : 2009-05-31  
Kernel Time = 0.546 = 59%  
User Time = 0.951 = 103%  
Process Time = 1.497 = 163%  
Global Time = 0.918 = 100%

16-threaded Exact search for '**incomprehensible\_misunderstanding**' into 889,537,624 bytes long file '4andabove\_Gamera.tar.2.sorted':

Allocating Master-Buffer 1536KB ... OK  
Kazahana: Dumped xgrams: 1  
Kazahana: Performance: 1,462 KB/clock  
Kazahana: Performance: Total/fread() clocks: 594/470  
Kazahana: Performance: I/O time, i.e. fread() time, is 79 percents  
Kazahana: Done.  
Timer 9.01 : Igor Pavlov : Public domain : 2009-05-31  
Kernel Time = 0.546 = 58%  
User Time = 0.889 = 95%  
Process Time = 1.435 = 154%  
Global Time = 0.930 = 100%

And to "verify" our watch with grep's:

'grep' 2.5.4 Exact search for '**ramjet**' into 889,537,624 bytes long file '4andabove\_Gamera.tar.2.sorted':

Kernel Time = 0.639 = 13%  
User Time = 4.196 = 86%  
Process Time = 4.836 = 99%  
Global Time = 4.847 = 100%

'grep' 2.5.4 Exact search for 'metal\_fatigue' into 889,537,624 bytes long file '4andabove\_Gamera.tar.2.sorted':

```
Kernel Time = 0.546 = 11%
User Time   = 4.040 = 88%
Process Time = 4.586 = 99%
Global Time = 4.587 = 100%
```

'grep' 2.5.4 Exact search for 'incomprehensible\_misunderstanding' into 889,537,624 bytes long file '4andabove\_Gamera.tar.2.sorted':

```
Kernel Time = 0.577 = 13%
User Time   = 3.806 = 86%
Process Time = 4.383 = 99%
Global Time = 4.399 = 100%
```

It surprises me how 'grep' doesn't get advantage of longer needles/patterns, anyway, Kazahana is 4.5:1 faster than 'grep'.

And one quick dummy stat:

For all three patterns below the total fread() time is in range 437..484 clocks (well, milliseconds):

'ramjet' is found at 1,011 KB/clock for entire task, but only 100% - 50% = 50% are spent in actual parsing and searching which equals 1,011 KB/clock / (50/100) = **2,022**

**KB/clock** L2 Multi-Threaded Gulliver performance.

'metal\_fatigue' is found at 1,235 KB/clock for entire task, but only 100% - 68% = 32% are spent in actual parsing and searching which equals 1,235 KB/clock / (32/100) =

**3,859 KB/clock** L2 Multi-Threaded Gulliver performance.

'incomprehensible\_misunderstanding' is found at 1,462 KB/clock for entire task, but only 100% - 79% = 21% are spent in actual parsing and searching which equals 1,462

KB/clock / (21/100) = **6,961 KB/clock** L2 Multi-Threaded Gulliver performance.

The bottom-line: Kazahana is bound to Main Memory Copying.

I have had too high hopes, speaking of total performance, the time which fread() consumes is a nasty break, meaning Kazahana is limited by I/O even with cached (by the OS) data. She is so fast that nowadays PCs with their slow memcpy() transfers ('Everest' says for Main Memory: 5446MB/s Read and 4015MB/s Copy on my laptop) are bottlenecks especially with MT-Gulliver used.

Seeing how XEONS do Main Memory Copy at 20000MB/s still gives me hope of getting 20000:4000 boost.

In my view, Kazahana is a forerunner (*a snowflower in a sunny day*) of more cool times, when RAM operations will be much faster and she will be able to bloom TRULY.

```
D:\_KAZE\GameraWiki\pediaWiktionary>"kazahana_r1-++fix+nowait_critical_nixFIX_wolfram+fixITER+EX+CS_fix_DEFINE_HEXADecad-Threads_IntelV12_SSE2_32bit.exe"
Kazahana, a superfast exact & wildcards & Levenshtein Distance (Wagner-Fischer) searcher, r. 1-++fix+nowait_critical_nixFIX_wolfram+fixITER+EX+CS_fix_DEFINE, copyleft Kaze 2014-Dec-04.
Usage: Kazahana [AtMostLevenshteinDistance][e] string textualfile MasterBufferSize
Note0: MasterBufferSize is in KB, consider 1024, 3072, 7168 or bigger. Two additional flags were mapped on this value: all dump
      lines (except fuzzy's) will have/lack pattern-source info when the number is even/odd respectively, see Examples #5 and #6.
Note1: There are three regimes: exact, wildcards and fuzzy searches. First two kick in when 3 parameters are given, fuzzy when 4.
Note2: What decides whether exact or wildcards? Of course presence of at least one wildcard. To see exact search see Example #4.
Note3: Exact search hits with 'Railgun_SekiReigan_Wolfram'.
Note4a: Incoming string is automatically lowercased for fuzzy searches i.e. they are case insensitive.
Note4b: Incoming string is NOT automatically lowercased for wildcards searches when MasterBufferSize ends in 0..4 i.e. they are case sensitive.
Note4c: Incoming string is automatically lowercased for wildcards searches when MasterBufferSize ends in 5..9 i.e. they are case insensitive.
Note5: Incoming string could be up to 26208/156 chars for Exact&Wildcard&ExhaustiveFuzzy/Fuzzy respectively.
Note5a: Since 2013-Nov-21 Levenshtein search exits not when the incoming line is bigger than 156 chars, now it just skips longer lines.
Note5b: Since 2013-Dec-05 Levenshtein search can be EXHAUSTIVE if LD is postfixed with 'e'.
Note6: Incoming textualfile could be bigger than 4GB.
Note7: Each line should end with [CR]LF, that is Windows or/and UNIX style.
Note8: The dump goes to Kazahana.txt file.
Note9a: Nine SLOW wildcards are available:
      wildcard '*' any character(s) or empty,
      wildcard '.' any ALPHA character(s) or empty,
      wildcard '_' any NON-ALPHA character(s) or empty,
      wildcard '@'/'#' any character {or empty}/{and not empty},
      wildcard '^'/'$' any ALPHA character {or empty}/{and not empty},
      wildcard '|'/'~' any NON-ALPHA character {or empty}/{and not empty}.
Note9b: Two FAST wildcards are available:
      wildcard '&' any character(s) or empty,
      wildcard '+' any character and not empty.
Note9c: Don't mix SLOW and FAST, the SLOW overrides the FAST, i.e. presence of at least one of the 9 wildcards cancels FAST mode.
Example1: E:>Kazahana 0 ramjet MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd 1536
Example2: E:>Kazahana 3 psychedllicize MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd 1536
Example3: E:>Kazahana "psyched^~~~~~ize^" MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd 1536
Example4: E:>Kazahana "metal fatigue" enwiki-20121201-pages-articles.xml 7168
Example5: E:>Kazahana "out~~~~~ize*" MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd 1536
      E:>type Kazahana.txt
      [out~~~~~ize*] outhyperbolize /MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd/
      [out~~~~~ize*] outsize /MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd/
      [out~~~~~ize*] outsize /MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd/
      [out~~~~~ize*] outsize /MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd/
      [out~~~~~ize*] outstrategize /MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd/
      [out~~~~~ize*] outtyrannize /MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd/
Example6: E:>Kazahana "out~~~~~ize*" MASAKARI_General-Purpose_Grade_English_Wordlist_r3_316423_words.wrd 1537
      E:>type Kazahana.txt
      outhyperbolize
      outsize
      outsize
      outstrategize
      outtyrannize
Example7: E:>Kazahana 2e edelvais MASAKARI_General-Purpose_Grade_English_Wordlist.wrd 1024
      E:>type Kazahana.txt
      bordelais
      bordelaise
      edelweiss
      edelweisses
      foredevised
      predellas
      psychedelicism
Info1: One second seems to have 1,000 clocks.
Info2: This CPU seems to be working at 2,829 MHz.
D:\_KAZE\GameraWiki\pediaWiktionary>
```

# Kazahana\_MokujIN\_Q9550s\_benchmark\_32vs64.txt:

```
D:\_KAZE\Gamerawikipediawiktionary>timer32.exe Kazahana_r1-++fix+nowait_critical_nixFIX_wolfram+fixITER+EX+CS_fix_DEFINE_HEXADECAD-Threads_IntelV12_SSE2_32bit 4e
"Silvestor Staloune" enwiki-20141008-pages-articles.xml 11263
Kazahana, a superfast exact & wilcards & Levenshtein Distance (Wagner-Fischer) searcher, r. 1-++fix+nowait_critical_nixFIX_wolfram+fixITER+EX+CS_fix_DEFINE, copyleft
Kaze 2014-Dec-04.
Pattern: Silvestor Staloune
omp_get_num_procs( ) = 4
omp_get_max_threads( ) = 4
Enforcing HEXADECAD i.e. hexadecuple-threads ...
Allocating Master-Buffer 11263KB ... OK
\; Speed: 00,000,002,129 bytes/clock; Traversed: 50,144,448,379 bytes
Kazahana: Total/Checked/Dumped xgrams: 800,855,553/342,059,464,575/2,106
Kazahana: Performance: 2 KB/clock
Kazahana: Performance: 33 xgrams/clock
Kazahana: Performance: Total/fread() clocks: 23,563,790/434,427
Kazahana: Performance: I/O time, i.e. fread() time, is 1 percents
Kazahana: Performance: RDTSC I/O time, i.e. fread() time, is 1,229,310,828,930 ticks
Kazahana: Done.

Kernel Time = 224.891 = 0%
User Time = 84620.215 = 359%
Process Time = 84845.106 = 360% Virtual Memory = 16 MB
Global Time = 23564.912 = 100% Physical Memory = 16 MB
```

D:\\_KAZE\Gamerawikipediawiktionary>dir

```
11/30/2014 08:49 PM 50,151,236,957 enwiki-20141008-pages-articles.xml
11/27/2014 04:35 PM 8,892,826,821 enwiki-20141008-pages-articles.xml.graffith
11/27/2014 04:35 PM 87 enwiki-20141008-pages-articles.xml.graffith.shal
11/27/2014 04:35 PM 78 enwiki-20141008-pages-articles.xml.shal
11/27/2014 04:35 PM 377,343,638 enwiktionary-20141004-pages-articles.xml.graffith
11/27/2014 04:35 PM 93 enwiktionary-20141004-pages-articles.xml.graffith.shal
11/27/2014 04:35 PM 84 enwiktionary-20141004-pages-articles.xml.shal
11/25/2014 04:22 PM 627,200 GRAFFITH_r2+_Graphein_2.3.0_Intel_12.1_32bit_768MB.exe
12/07/2014 11:25 AM 1,064,420 Kazahana.txt
12/04/2014 02:27 PM 489,984 Kazahana_r1-++fix+nowait_critical_nixFIX_wolfram+fixITER+EX+CS_fix_DEFINE_HEXADECAD-Threads_IntelV12_SSE2_32bit.exe
12/04/2014 02:27 PM 581,120 Kazahana_r1-++fix+nowait_critical_nixFIX_wolfram+fixITER+EX+CS_fix_DEFINE_HEXADECAD-Threads_IntelV12_SSE2_64bit.exe
11/27/2014 04:35 PM 4,096 timer32.exe
11/30/2014 08:27 AM 55,532,553,655 _Gamera_r21.tar.LBL.txt
11/30/2014 01:16 PM 11,045,905,373 _Gamera_r21.tar.LBL.txt.graffith
11/30/2014 03:01 PM 76 _Gamera_r21.tar.LBL.txt.graffith.shal
11/30/2014 02:12 PM 67 _Gamera_r21.tar.LBL.txt.shal
11/27/2014 04:35 PM 11,801,663,755 _Gamera_r21.tar.txt.graffith
11/27/2014 04:35 PM 72 _Gamera_r21.tar.txt.graffith.shal
11/27/2014 04:35 PM 63 _Gamera_r21.tar.txt.shal
```

```
D:\_KAZE\Gamerawikipediawiktionary>timer32.exe Kazahana_r1-++fix+nowait_critical_nixFIX_wolfram+fixITER+EX+CS_fix_DEFINE_HEXADECAD-Threads_IntelV12_SSE2_64bit 4e
"Silvestor Staloune" enwiki-20141008-pages-articles.xml 11263
Kazahana, a superfast exact & wilcards & Levenshtein Distance (Wagner-Fischer) searcher, r. 1-++fix+nowait_critical_nixFIX_wolfram+fixITER+EX+CS_fix_DEFINE, copyleft
Kaze 2014-Dec-04.
Pattern: Silvestor Staloune
omp_get_num_procs( ) = 4
omp_get_max_threads( ) = 4
Enforcing HEXADECAD i.e. hexadecuple-threads ...
Allocating Master-Buffer 11263KB ... OK
\; Speed: 00,000,001,972 bytes/clock; Traversed: 50,144,448,379 bytes
Kazahana: Total/Checked/Dumped xgrams: 800,855,553/342,059,464,575/2,106
Kazahana: Performance: 1 KB/clock
Kazahana: Performance: 31 xgrams/clock
Kazahana: Performance: Total/fread() clocks: 25,432,596/695,557
Kazahana: Performance: I/O time, i.e. fread() time, is 2 percents
Kazahana: Performance: RDTSC I/O time, i.e. fread() time, is 1,968,171,428,160 ticks
Kazahana: Done.

Kernel Time = 274.015 = 1%
User Time = 93211.346 = 366%
Process Time = 93485.362 = 367% Virtual Memory = 17 MB
Global Time = 25433.485 = 100% Physical Memory = 16 MB
```

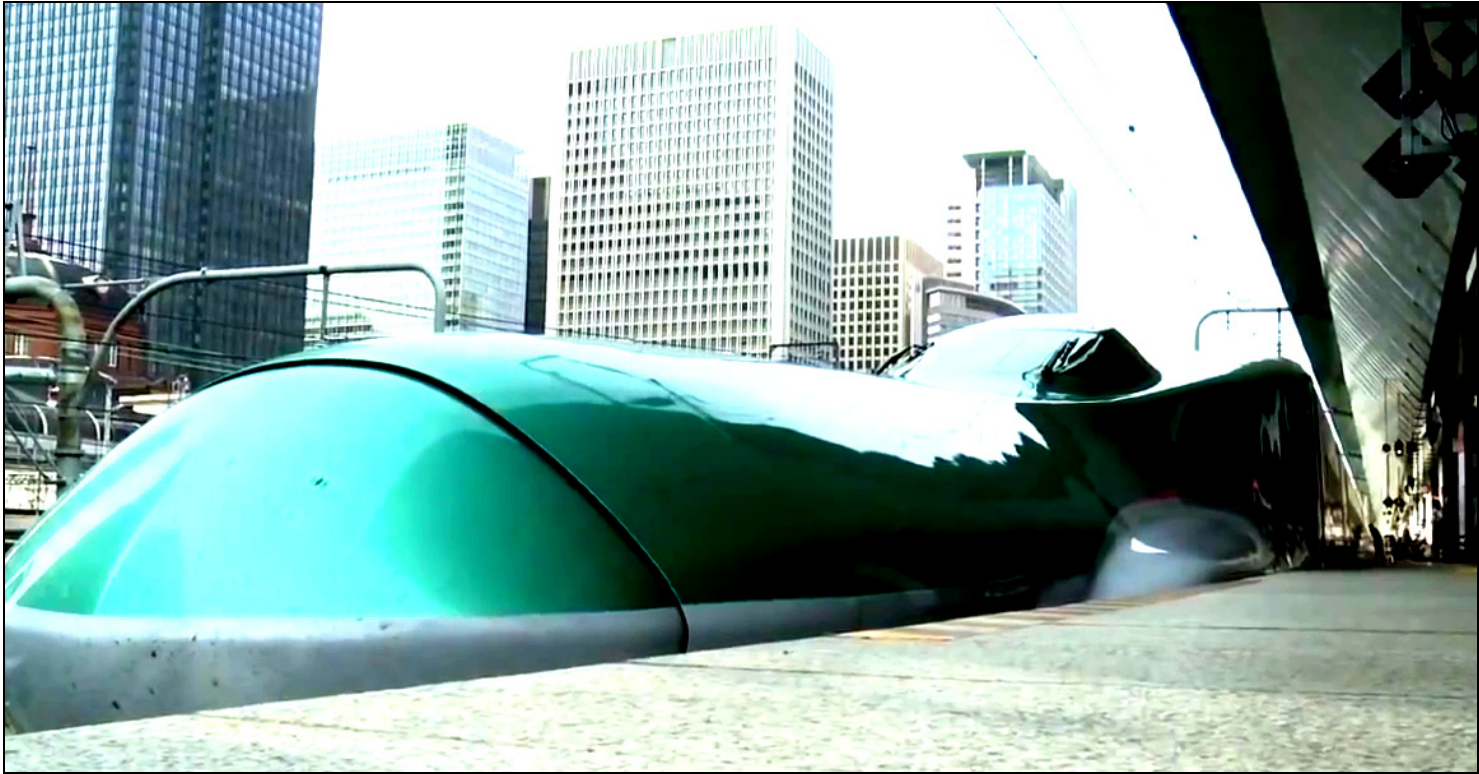
```
D:\_KAZE\Gamerawikipediawiktionary>timer32 "kazahana_2014-Dec-04\MokujIN_r5+_16-Threads_IntelV12_64bit_03.exe" 2 16777216 /stats
MokujIN, Multiplication of INtegers, an OpenMP (multi-threaded) string multiplier, 16 threads enforced, written by Kaze, 2012-Nov-16, revision 5fix+.
omp_get_num_procs( ) = 4
omp_get_max_threads( ) = 4
Multiplying performance for operands 00,000,001 digits long (footprint: ~000,000KB, checksum: 64b6,5527): 1 MokujINS i.e. digits per second.
Multiplying performance for operands 00,000,001 digits long (footprint: ~000,000KB, checksum: bfa2,d466): 1 MokujINS i.e. digits per second.
Multiplying performance for operands 00,000,002 digits long (footprint: ~000,000KB, checksum: 5fec,d95b): 4 MokujINS i.e. digits per second.
Multiplying performance for operands 00,000,003 digits long (footprint: ~000,000KB, checksum: 5366,0fd0): 9 MokujINS i.e. digits per second.
Multiplying performance for operands 00,000,005 digits long (footprint: ~000,000KB, checksum: 65ff,ca3d): 25 MokujINS i.e. digits per second.
Multiplying performance for operands 00,000,010 digits long (footprint: ~000,000KB, checksum: 27c1,8b40): 100 MokujINS i.e. digits per second.
Multiplying performance for operands 00,000,020 digits long (footprint: ~000,000KB, checksum: 2c6c,3fab): 400 MokujINS i.e. digits per second.
Multiplying performance for operands 00,000,039 digits long (footprint: ~000,001KB, checksum: cd36,3818): 1,521 MokujINS i.e. digits per second.
Multiplying performance for operands 00,000,078 digits long (footprint: ~000,002KB, checksum: 0344,bebb): 6,084 MokujINS i.e. digits per second.
Multiplying performance for operands 00,000,155 digits long (footprint: ~000,004KB, checksum: 192b,ccd2): 24,025 MokujINS i.e. digits per second.
Multiplying performance for operands 00,000,309 digits long (footprint: ~000,009KB, checksum: 45bf,a8a8): 95,481 MokujINS i.e. digits per second.
Multiplying performance for operands 00,000,617 digits long (footprint: ~000,019KB, checksum: c4d7,f586): 380,689 MokujINS i.e. digits per second.
Multiplying performance for operands 00,001,234 digits long (footprint: ~000,038KB, checksum: e624,05dc): 1,522,756 MokujINS i.e. digits per second.
Multiplying performance for operands 00,002,467 digits long (footprint: ~000,077KB, checksum: 3e13,6779): 6,086,089 MokujINS i.e. digits per second.
Multiplying performance for operands 00,004,933 digits long (footprint: ~000,154KB, checksum: 6cc3,3a31): 24,334,489 MokujINS i.e. digits per second.
Multiplying performance for operands 00,009,865 digits long (footprint: ~000,308KB, checksum: 5153,c152): 97,318,225 MokujINS i.e. digits per second.
Multiplying performance for operands 00,019,729 digits long (footprint: ~000,616KB, checksum: 0dbc,7708): 389,233,441 MokujINS i.e. digits per second.
```

Multiplying performance for operands 00,039,457 digits long (footprint: ~001,233KB, checksum: 8a6a,6290): 389,213,712 MokujINS i.e. digits per second.  
Multiplying performance for operands 00,078,914 digits long (footprint: ~002,466KB, checksum: ec28,02e2): 444,815,671 MokujINS i.e. digits per second.  
Multiplying performance for operands 00,157,827 digits long (footprint: ~004,932KB, checksum: 4f7a,37cd): 429,471,757 MokujINS i.e. digits per second.  
Multiplying performance for operands 00,315,653 digits long (footprint: ~009,864KB, checksum: 24d1,7bb1): 429,469,036 MokujINS i.e. digits per second.  
Multiplying performance for operands 00,631,306 digits long (footprint: ~019,728KB, checksum: 84f8,a441): 432,263,845 MokujINS i.e. digits per second.  
Multiplying performance for operands 01,262,612 digits long (footprint: ~039,456KB, checksum: 6eff,9dd7): 431,444,942 MokujINS i.e. digits per second.  
Multiplying performance for operands 02,525,223 digits long (footprint: ~078,913KB, checksum: ald4,99ab): **431,123,737 MokujINS** i.e. digits per second.  
Dumping the result to 'MokujIN.txt' ... OK  
Total Time: 19,718 second(s).  
Kernel Time = 0.686 = 0%  
User Time = 76840.289 = 389%  
Process Time = 76840.975 = 389% Virtual Memory = 475 MB  
Global Time = 19717.633 = 100% Physical Memory = 90 MB

D:\\_KAZE\GamerawikipediaWiktionary>dir MokujIN.txt  
12/08/2014 12:44 PM 5,050,448 MokujIN.txt

D:\\_KAZE\GamerawikipediaWiktionary>7za.exe b  
7-Zip (A) 9.20 Copyright (c) 1999-2010 Igor Pavlov 2010-11-18  
RAM size: 2047 MB, # CPU hardware threads: 4  
RAM usage: 850 MB, # Benchmark threads: 4  
Dict Compressing | Decompressing  
Speed Usage R/U Rating Speed Usage R/U Rating  
KB/s % MIPS MIPS KB/s % MIPS MIPS  
22: 8748 343 2482 8510 | 126795 398 2875 11440  
23: 8467 350 2467 8627 | 122918 396 2839 11248  
24: 8300 359 2486 8925 | 120498 395 2827 11179  
25: 7838 343 2611 8949 | 115334 384 2827 10846  
-----  
Avr: 349 2512 8753 | 393 2842 11178  
Tot: 371 2677 **9965**

On the shot below, E5 aka 'Hayabusa' with its long nose able to 'pierce' tunnels in order to suppress airwaves.



Machinely yours Kaze,  
2014-Dec-11