SAVINGS TIME EXECUTION PRIMA NUMBERS GENERATOR USING BIT-ARRAY STRUCTURE

Letnan Kolonel Elektronika Imat Rakhmat Hidayat, S.T., M.Eng

1 Kepala Bagian Teknologi dan Informasi Pasat Pengendalian Operasi Mabes TNI Cilangkap Jakarta Timur, Indonesia
* Corresponding author E-mail: imat96@gmail.com

Abstract—Prime number in growth computer science of number theory and very need to yield an tool which can yield an hardware storey level effectiveness use efficiency and Existing Tools can be used to awaken regular prime number sequence pattern, structure bit-array represent containing subdividing variables method of data aggregate with every data element which have type of equal, and also can be used in moth-balls the yielded number sequence. Prime number very useful to be applied by as bases from algorithm kriptografi key public creation, hash table, best algorithm if applied hence is prime number in order to can minimize collision (collisions) will happen, in determining pattern sequence of prime number which size measure is very big is not an work easy to, so that become problems which must be searched by the way of quickest to yield sequence of prime number which size measure is very big. Serial use of prosesor in seeking sequence prime number which size measure is very big less be efficient remember needing of computing time which long enough, so also plural use prosesor in seeking sequence of prime number will concerning to price problem and require software newly. So that by using generator of prime number use structure bit-array expected by difficulty in searching pattern sequence of prime number can be overcome though without using plural processor even if, as well as time complexity minimization can accessed. Execution time savings gained from the research seen from the research data, using the algorithm on the input Atkins 676,999,999. 4235747.00 execution takes seconds. While the algorithm by using an array of input bits 676,999,999. 13955.00 execution takes seconds. So that there is a difference of execution time for 4221792.00 seconds.

1. Introduction

The study of prime numbers has become part of one branch of mathematics, namely Number Theory[1]. Prime problem has become one of the subject of research conducted today [7]. Prime numbers are positive integers greater than one who has the right only two factors [2], namely the number 1 (one) and the number itself [3]. Numbers other than primes that have more than two factors and in addition to the number 1 (one) known as composite numbers [4]. Number 1 (one) is a special case, not including the prime or composite [17]. Although the numbers 1 (one) was once considered a prime (Goldbach 1742; Lehmer 1909; Lehmer 1914; Hardy and Wright 1979, pp. 11; Gardner 1984, pp. 86-87; Sloane and Plouffe 1995, pp. 33; Hardy 1999 , pp. 46), required special handling in many definitions and applications that include prime numbers, so the number 1 (one) is usually placed in a special class. Primalitas is a state in which a prime number, including [13]. The use of prime numbers is a serious problem in the development of computer science and number theory[5].

Prime number generator is a tool [6], which can generate the sequence patterns of regular prime numbers, bit-array structure is a grouping method of variables that contain data sets with each data element of the same type used in the storage of generated sequence numbers [7]. Computers today are digital computers [8], these computers only understand numbers 0 dan 1, all numbers can be represented by using the Bit (Binary digit) [20]. Prime numbers are useful for purposes such as may be applicable as the basis of the creation of public-key cryptography algorithm [9], an important part of the cryptographic system is factoring large numbers into prime numbers factor [10], there are still problems of determining the sequence of prime numbers of scale very large [11]. Another benefit of using prime numbers can be used in the hash table [12], hash table algorithm is best used when initialized with a prime number in order to min-
imize collisions (collisions) is going to happen [18], in determining the sequence patterns of prime numbers is very large sekalanya is not a job easy [19], so this becomes a problem to be solved [14].

Thus by using bit-array structure is expected difficulties in the search for sequence patterns of prime numbers can be solved even without using multiple processors though [15]. So the execution time savings can be made.

2. Algorithm

Judging from the origins of his own he said the word algorithm has a strange history [16]. People just find the word algorithm, which means the process of arithmetic (calculation) using arabic numbers: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 (Webster's New Word Dict, 1957). The linguist tried to find the origin of this word but the results are less satisfactory [20]. Finally, the historians of mathematics found the origin of the word is derived from the name of the author of the famous arab Abu Ja'far Muhammad ibn Musa al-Khwarizmi. Al-Western Khwarizmi be read Algorism. Al-Khwarizmi wrote a book entitled The Book of Algebra WalMuqabala which means "The Book of restoration and reduction" (The Book of restoration and reduction). From the title of the book we also get the root word "Algebra" (Algebra).

In medieval abacus calculation by using algorism, so algorithm better known as algiros (painfull) + arithmos (number). Change the word of the Algorithm Algorithm arises because the word is often confused with Algorithm Arithmetic [21], so the suffix became-sm-THM [24]. Because calculations with arabic numbers have become bisaa things, the words gradually fade algorithm is used as a method of calculation (computing) in general [22], thereby losing its original meaning [23]. In Indonesian, the word Algorithm is absorbed into algorithm.

The algorithm is the basic of all computer programming. Algorithm modern meaning [25]: as a recipe, process, method, technique, procedure, routine. According to Donald E Knuth algorithm must meet the following requirements:

1. **Finiteness**
   Algorithm to end (terminate) after doing some step process.

2. **Determinism**
   Every step of the algorithm must be appropriately defined and do not cause double meaning (ambiguous). Because it is actually the most appropriate way to write the algorithm is to use formal language (computer programming language).

3. **Input**
   Each algorithm requires as input data to be processed. The algorithm does not require any input really not very useful because the number of cases that can be solved also limited.

4. **Output**
   Each algorithm provides one or more of the output.

5. **Effectiveness**
   Algorithm steps done in a reasonable time.

As the basis of computer programming [26], the algorithm describes the order of the steps necessary for solving problems (problem solving), which has a characteristic as follows:

1. Always have a termination/final step.
2. Each step is clearly stated and firmly.
3. Each simple step so that its performance in relation to the efficient time/make sense.
4. Gives the result (output), perhaps with one or no input

2.1. Algoritma Sieve Of Eratosthenes

Sieve of Eratosthenes is a classical algorithm for finding all prime numbers up to a specified N. Start with an array of integers that have not been stricken from 2 to N. The first integer that is not crossed 2, is the first prime numbers. Eliminate all the multiples of these primes. Repeat on the next integer that has not crossed out. Sieve of Eratosthenes algorithm is used as the basis for the writer to make a prime generator using bit array structure. Sequence of steps of the algorithm Sieve of Eratosthenes can be seen in the following explanation:

1. First of all, make a list of numbers from 2 up to a largest number which we denote by n.
2. Elimination or remove from the list of all numbers multiples of 2.
3. The next number not yet eliminated is of prime numbers.
4. Elimination or remove from the list of all numbers that are multiples of the numbers in the previous step.
5. Repeat step 3 and step 4 until the number is greater than the number remaining after step 5 is a prime number.

This algorithm has time complexity O (n log log n). Version of this algorithm with the optimization process, such as wheel factorization, has a time complexity O (n). For example, here is an array in the first place: 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27.

Since 2 has not crossed out, then the prime number 2 is our first. We crossed out all the multiples of 2, namely 4, 6, 8, 10, 12, onwards 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27.

Next integer that has not crossed out is 3, then 3 is prime and we crossed out all the multiples of 3, such as 6, 9, 12, onwards.

Next integer that has not crossed out is 5, then 5 is prime and we crossed out all the multiples of 5. The only number that's out of this range is 25.

So we get the next prime number is 7, 11, 13, 17, 19, and 23.
Listing in pseudocode is:

```pseudocode
// set limits
limit ← 1000000
// Set all the numbers with true
is_prime (i) ← true, i [2, limit]
for n in [2, √ limit]:
    is_prime if (n):
        is_prime (i) ← false,
        i (n², n² + n, n² +2 n, ..., limit)
for n in [2, limit]:
    is_prime if (n): print n
```

More simply can be seen in pseudocode as follows:

```pseudocode
limit = 1000000
sieve $ = string "P" with a long line
prime = 2
repeat while prime² < limit
    set of characters in the index of each
    multiples of prime numbers (numbers outside the * 1)
    in sieve $ becomes "n"
    prime = index of each "p" in the sieve
end repeat
```

Pseudocode above if translated into a flowchart shape can be seen in the Figure below:

![Flowchart Program Algorithm of Sieve Of Eratosthenes](image)

**Fig. 1:** Flowchart Program Algorithm of Sieve Of Eratosthenes
3. Results and Discussion

3.1. Pseudocode Sieve Of Eratosthenes

Algorithm used in determining deret of prime number to n of number determined by is Algorithm of Sieve Of Eratosthenes, presented in Pseudocode hereunder:

```c
// Set limit
limit ← 1000000
// Set all the numbers with true
is_prime (i) ← true, i [2, limit]
for n in [2, √ limit]:
    is_prime if (n):
        is_prime (i) ← false,
        i (n^2, n^2 + n, n^2 +2 n, ..., limit)
for n in [2, limit]:
    is_prime if (n); print n
```

Or more as simple as:

```c
limit = 1000000
sieve $ = string "P" with a long line
prime = 2
repeat while prime2 < limit
    set of characters in the index of each
    multiples of prime numbers (numbers outside the * 1)
    in sieve $ becomes "n"
    prime = index of each "p" in the sieve
$ end repeat
```

Writer translate pseudocode of above becoming a program in language C

3.2. Pseudocode Sieve of Atkins

Algorithm of Sieve of Atkins to determine sequence of prime number to n of number determined to be presented in Pseudocode hereunder:

```c
limit ← 1.000.000
prima(i) ← false, i [5, limit]
for (x, y) in [1, √batas] × [1, √limit]:
    n ← 4x²+y²
    if (n ≤ limit) (n mod 12 = 1 n mod 12 = 5):
        prima(n) ← ¬prima(n)
    n ← 3x²+y²
    if (n ≤ limit) (n mod 12 = 7):
        prima(n) ← ¬prima(n)
    n ← 3x²y²
    if (x > y) (n ≤ limit) (n mod 12 = 11):
        prima(n) ← ¬prima(n)
    for n in [5, √batas]:
        if prima(n):
            prima(k) ← false, k [n², 2n², 3n²,..., limit]
print 2, 3
for n in [5, batas]:
    if prima(n): print n
```

Writer translate pseudocode of above becoming a program in Language C

3.3. Execute Program

Executing of Program conducted by using tool Dev-C++, Program using bent of array hence Algorithm of Sieve Of Eratosthenes, and Program generator of prime number which hence algorithm of Sieve of Atkin, executed by using computer of single prosesor with specification ( Prosessor Intel Pentium III 601 MHZ, Louvre 256 MB, Hardisk 20 GB, Operating system of Professional Microsoft windows XP of Service Pack 2. Executing generator of prime number by using structure of bent of array using algorithm of Sieve of Eratosthenes use tool Dev-C++, visible at Figure 2.
Appearance after program in compiling visible at Figure 3. Seen at appearance menu there is dialogue box Compil Progress seen at Compiler contain Default compiler, Status contain Done, Errors contain 0 (Zero), and Warning contain 0 (Zero), this information is menunjukan that program have there no mistake.

Program in Compiling and have there no mistake, hereinafter program in Run. Appearance after program in Run seen at Figure 4. In example of this case is number input number input 100.000, seen seen by a time depth execute during 4.00 second.
Program generator of prime number without structure of beet of array use algorithm of Sieve of Atkins checked hence is tool Dev-C++, visible at Figure 4.

Program generator of prime number without structure of beet of array use algorithm of Sieve of Atkins checked hence is tool Dev-C++, visible at Figure 5.

Appearance after program executed later then in compiling and visible run at Figure 6. At menu of appearance seen by a screen emerge comand for user of input of number to be searched by deret of its prime number

Appearance of Menu after "Input Assess N:" filled with number determined by user, in example of hereunder input of number which input of equal to 100.000. After in run seen by a time depth execute during 30 second, visible at Figure 7

3.4. Result of Comparison

In execution of this attempt, is conducted by analysis of attempt start from input 10.000 up to 676.999.999, executed by using computer of single prosesor with specification ( Prosessor Intel Pentium III 601 MHZ, RAM 256 MB, Hardisk 20 GB, Operating system of Professional Microsoft windows XP,Service Pack 2). Is later then calculated by time executed, result of attempt second of visible program at tables in the following is
| No. | Input       | Time executed | Time executed | Bold |
|-----|-------------|---------------|---------------|------|
|     |             | Algoritma Bit-Array (Second) | Algoritma Atkins (Second) |      |
| 1   | 10,000      | 0.00          | 0.00          | -    |
| 2   | 40,000      | 2.00          | 6.00          | -    |
| 3   | 50,000      | 1.00          | 9.00          | -    |
| 4   | 70,000      | 3.00          | 16.00         | -    |
| 5   | 80,000      | 4.00          | 20.00         | -    |
| 6   | 100,000     | 4.00          | 25.00         | -    |
| 7   | 130,000     | 4.00          | 30.00         | -    |
| 8   | 140,000     | 6.00          | 57.00         | -    |
| 9   | 200,000     | 10.00         | 111.00        | -    |
| 10  | 240,000     | 8.00          | 156.00        | -    |
| 11  | 250,000     | 8.00          | 169.00        | -    |
| 12  | 300,000     | 14.00         | 237.00        | -    |
| 13  | 340,000     | 11.00         | 307.00        | -    |
| 14  | 350,000     | 12.00         | 319.00        | -    |
| 15  | 360,000     | 11.00         | 337.00        | -    |
| 16  | 370,000     | 12.00         | 353.00        | -    |
| 17  | 400,000     | 14.00         | 411.00        | -    |
| 18  | 500,000     | 16.00         | 627.00        | -    |
| 19  | 1,000,000   | 30.00         | 2358.00       | -    |
| 20  | 2,000,000   | 64.00         | 8931.00       | -    |
| 21  | 5,000,000   | 138.00        | 19116.00      | -    |
| 22  | 10,000,000  | 266.00        | 38232.00      | -    |
| 23  | 100,000,000 | 2235.00       | 392421.00     | -    |
| 24  | 500,000,000 | 10681.00      | 1912105.00    | -    |
| 25  | 676,000,000 | 13945.00      | 3843326.00    | -    |
| 26  | 676,999,999 | 13955.00      | 4235747.00    | -    |

From result data of is conducted attempt, can be projected by into form of the following graph

**Fig. 8:** Graph of Comparison of Time Executed

The results of research, using the algorithm on the input Atkins 676,999,999. 4235747.00 execution takes seconds. While the algorithm by using an array of input bits 676,999,999. 13955.00 execution takes seconds. So that there is a difference of execution time for 4221792.00 seconds.

### 4. Conclusion

Conclusion which can be taken away from by result of attempt and analysis is:

- Structure of Bit-Array represent method subdividing of containing variable-variable of data aggregate with each; every its data element have is type to of equal, used in depository of sequence of number yielded
- By using structure of bit-array of difficulty in searching pattern of sequence of prime number which the scorpion very big can be overcome by using single processor, and can quicken time accessed
- Execution time savings gained from the research seen from the results of research, using the algorithm on the input Atkins 676,999,999. 4235747.00 execution takes seconds. While the algorithm by using an array of input bits 676,999,999. 13955.00 execution takes seconds. So that there is a difference of execution time for 4221792.00 seconds.

Suggestion which can be given for the research hereinafter is suggested for the next research make an generator of prime number by using structure of array beef, and the Algorithm used by Algorithm of Sieve of atkins.
Acknowledgement

This research was held in collaboration between researchers and Pusdalops Mabes TNI. All funding from research implementation to publication in the journal Multica Science and Technology (MST) comes from the Pusdalops Mabes TNI. Researchers would like to thank Pusdalops Mabes TNI and also Gadjah Mada University, who have contributed to contribute to this research. Thanks to my wife, Yayang Febry Argusna, S.Sos, who has been an encouragement for researchers to complete this research.

References

[1] Abdul Kadir & Heriyanto. 2005. Algoritma Pemrograman Menggunakan C++, Andi.
[2] Abdul Kadir. 2004. Panduan Pemrograman Visual C++, Andi.
[3] Abdul Kadir. 1992. Pemrograman Dasar Turbo C Untuk IBM PC, Andi offset.
[4] A. Rida Ismu Windyarto. 2007. Diktat Kuliah Analisis Algoritma. Program Pascasarjana Teknik Elektro, Universitas Gadjah Mada.
[5] Bambang Hariyanto. 2000. Buku Teks Ilmu Komputer Struktur Data, Informatika
[6] Bruno, R. Preiss. 1999. Data Structures and Algorithms with object-oriented design patterns in C++, John Wiley & Sons.
[7] David Wells. 2005 Prime Numbers: The Most Mysterious Figures in Math. John Wiley & Sons. Inc., ISBN 0-471-46234-9.
[8] Eko Budi Purwanto. 2008. Perancangan & Analisis Algoritma, Graha Ilmu.
[9] Gok Asido Haro. (2007, Mei 06). Re: Algoritma Pencarian Bilangan Prima.
[10] [Online]. Available: if15072@students.if.itb.ac.id [2007, Mei 06].
[11] Gudono. 1995. Pemrograman C dan C++, Andi offset
[12] Hanagra Abidianto. (2009, Januari 04). Re: Bilangan Prima dan Aplikasinya dalam Bidang Informatika. [Online]. Available: that.kid.is.sherlock@gmail.com [2009, Januari 04].
[13] Jansen. (2007, Februari 27). Re: Studi sejarah dan perkembangan bilangan prima.
[14] [Online]. Available: if16028@students.if.itb.ac.id [2007, Februari 27].
[15] John R. Hubbard. 2000. Data Structures With C++, McGraw-hill Companies.
[16] Mehta, Dp. and Sahni, S. 2005. Handbook of DATA STRUCTURES and APPLICATIONS, Chapman & Hall/CRC Computer and information Science.
[17] Mc Millan, M. 2007. Data Structures and Algorithms Using C#, Cambridge. University Press.
[18] Munir, Rinaldi. 2008. Diktat Kuliah IF209 Struktur Diskrit. Bandung : Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung. Hal. V-19 – V-21.
[19] Rinaldi munir & Leoni lidya. 2002. Algoritma dan Pemrograman dalam Bahasa Pascal dan C, Informatika.
[20] Hamonangan Situmorang. (2009, Mei 18). Algoritma dan Struktur Data, telecom.ee.itb.ac.id/~monang/if2031/Array_dalam_Bahasa C.
[21] Wahyuni. 2004. Sistem Berkas, Andi.
[22] http://www.troubleshooters.com/codecom/primenumbers/primenumbers.htm
[23] http://en.wikipedia.org/wiki/Bit_array
[24] http://stackoverflow.com/questions/231381/is-this-prime-generator-inefficient-c [25] http://www.mti.ugm.ac.id/~nadyIX%20Struktur%20Pointer.pdf
[26] http://www.google.co.id/#hl=id&q=pengertian+pointer&meta=cr%3DcountryID&aq=f&oq=pengertian+pointer&fp=b410d05dd5c338e1
[27] http://downloads.ziddu.com/downloadfile/4881612/POINTER.doc.html