

ДАРААЛСАН АНХНЫ ТООНУУДЫГ ОЛОХ БОДЛОГЫН ШИНЖИЛГЭЭ БА ШИЙДЭЛ

Д.Бүрэн-Арвижих
Ховд их сургууль
burenarvijikh@gmail.com

Хураангуй

Мэдлэгт суурилсан, хүмүүнлэг, ардчилсан нийгэмд ажиллаж амьдрах мэдээллийн технологийн мэргэжилтэн бүр хариуцлагатай, ухамсартай, бүтээлчээр амьдрах, зах зээлийн чөлөөт өрсөлдөөнд шударгаар ойролцох, аливаа үзэгдлийг задлан шинжлэх, нэгтгэн дүгнэх, бодит хэрэгцээ шаардлагатай тулгардаг бөгөөд мэдээлэл зүйн боловсрол эзэмших, тэр дундаа бодлогыг программчлах үйл ажиллагаанд сурах явцдаа өөрийн бие даасан бүтээлч үйл ажиллагаагаар өмнө эзэмшсэн олон төрлийн мэдлэг, чадвар дээрээ тулгуурлан шинэ мэдлэг олж авах, хийж гүйцэтгэх, аливаа асуудлын харилцан уялдаа, холбоо хамаарлыг таньж мэдэх үйлийг хөгжүүлэхийн зэрэгцээ аливаа нэгэн бодлогын олон төрлийн шийдэл, алгоритмуудыг харьцуулах, шинжлэх, хамгийн оновчтойг нь сонгох чадварыг эзэмших нь ихээхэн чухал юм.

Түлхүүр үг: Алгоритм, тоонуудын олонлог, зохиомол тоо, программ, код

Оршил

Ерөнхий боловсролын сургууль болон их дээд сургуулийн хөтөлбөрт тусгагдсан мэдээлэл зүй, алгоритм программчлалын үндэс болон бодлого бодох арга хичээлийн хүрээнд авч үзэгдэх бодлого, алгоритм дотор анхны тоо, түүнтэй холбогдох бодлого ихээхэн хувийг эзлэхийн зэрэгцээгээр олон төрлийн бодлогын программ зохиох явцад энэ төрлийн бодлогын шийдэл нь үндэс суурь болон ашиглагддаг.

Иймд энэхүү өгүүлэлээр анхны тоо түүнтэй холбогдох нэгэн бодлого, түүнийг бодох алгоритмуудын тухай авч үзэж, баримт бичгийн судлагаа болон задлан шинжлэсний үндсэн дээр уг бодлогын программчлалын паскаль хэл дээр бичигдсэн хоёр кодыг сайжруулан, боловсруулах зорилго тавилаа.

Мэдээлэл зүйн бодлого анх тавигдсан үеэсээ эхлэн цаг хугацааны явцад олон арга хувилбараар шийдэгдэн, улам төгөлдөржин хамгийн оптимал шийдлээр шийдэгдсэн ч түүний авч үзсэн орчин, шийдэхэд ашигласан ойлголт, хэрэглүүр зэргээс хамааран олон арван хувилбараар шийдэгдэж байдаг онцлогтой билээ.

Иймд Эртний Грекийн математикч, астрономич, газарзүйч, яруу найрагч **Eratosphenes Kirensky** (МЭӨ 276 – МЭӨ 194) анх томъёолсон Анхны тоо олох дээр дурдагдсан алгоритмыг хожим 1975 онд

Английн эрдэмтэн David Turner хамгийн сонгодог хэлбэрээр нь томъёолжээ.

Бид ч мөн энэ зүй тогтлын дагуу тухайн алгоритмыг программчлах өөрсдийн хувилбарыг дэвшүүлж байна.

Бодлогын тавил: n тоо өгөгдсөн бол 1 - ээс өгөгдсөн n тоо хүртлэх тоонуудаас бүх анхны тоог ол.

Үндсэн хэсэг:

Бодолт: Энэхүү бодлогыг хэрхэн шийдэх вэ? гэсэн асуулттай холбогдон Эратосфений тор гэж нэрлэгдсэн алгоритм яригддаг. Энэхүү алгоритмын үндсэн санаа нь “нэг ба өөрөөсөө өөр натурал тоон хуваагчгүй натурал тоог анхны тоо гэнэ” гэсэн тодорхойлолт дээр тулгуурладаг бөгөөд өгөгдсөн завсрын бүх тоог агуулсан тоон олонлогийг байгуулсны дараа уг олонлогоос зохиомол тоонуудыг нь дэс дараалан зайлуулахад бүх анхны тоонууд шүүгдэн хоцрох ёстой гэсэн зарчмаар хэрэгждэг.

Алгоритмыг шийдэх явцад

- Өгөгдсөн тоон завсрын бүх тоог агуулсан олонлогийг хэрхэн байгуулах
- Олонлогийн элементүүдийг хэрхэн шалгах
 - Олонлогийн ээлжит элементийг сонгон анхны тоо мөн эсэхийг шалгах
 - Шалгалтыг гүйцэтгэх үед сонгогдсон элементийн хуваагчдыг хэрхэн сонгох
- Олонлогоос зохиомол тоонуудыг хэрхэн зайлуулах
- Шүүгдэж хоцорсон анхны тоонуудыг хэрхэн хэвлэх вэ? гэсэн асуудлуудыг шийдэх шаардлагатай.

Судалгааны үр дүн. Эдгээр асуудлуудыг шийдэн уг бодлогыг Паскаль хэл дээр программчилсан нэгэн хувилбар кодыг авч үзье.

Код 1. Программчлалын Паскаль хэл, Олонлог ашигласан

```
Program Eratosphen01;
Uses Wincrt;
Var
  M : Set of Byte;
  i, k, n : Integer;
Begin
  Writeln('Завсрын хэмжээг (255 хүртэл) оруулна уу');
  Readln(n);
```

$M := [2..n]$; { Өгөгдсөн завсрын бүх тоог агуулсан олонлог байгуулах }

For $k := 2$ to $n \div 2$ do { Олонлогийн бүх тоонуудын хуваагчдыг түүх }

For $i := 2$ to n do { Ээлжит тоог сонгох }

If $(i \bmod k = 0)$ and $(k < i)$ Then { i тоо өөрөөсөө ялгаатай k хуваагчид хуваагдах эсэхийг шалгах }

$M := M - [i]$; { i зохиомол тоог олонлогоос хасах }

For $i := 2$ to n do { Анхны өгөгдсөн завсрын бүх тоог сонгох }

If i in M Then { байгуулсан олонлогт ээлжит i тоо үлдсэн эсэхийг шалгах }

Write(i , ' '); { анхны тоо i - г хэвлэх }

End. (Чойжоованчиг.Л, Цэдэвсүрэн.Д, & Хатанхангай.Ц, 2005), (Гармаа.Д, 2009)

Дээрх программаас харахад уг алгоритмын ажиллах чадварыг бууруулж байгаа явдал нь n тоог 255 – аас бага буюу тэнцүү байхаар оруулан өгч байгаа явдал юм. Авч үзэж буй Эратосфений тор алгоритм нь учир дутагдалтайд биш, харин ихэнх ном сурах бичигт “Олонлог” сэдвийг судлах явцад жишээ бодлого байдлаар уг алгоритмыг авч үзсэн явдал бөгөөд олонлог нь харьцангуй хязгаарлагдмал өөрөөр хэлбэл 256 ширхэг элемент агуулах боломжтой байдаг явдал юм.

Иймээс уг алгоритмыг бүрэн боломжоор нь ашиглахын тулд массив болон бусад илүү олон элемент агуулах боломжтой өгөгдлийн нийлмэл төрлийг ашиглан шийдэх шаардлагатай гэж үзээд массив ашиглан шийдсэн алгоритмын хийсвэр кодыг авч үзье.

Код 2. Эратосфений тор алгоритмын хийсвэр код

Оролт: Бүхэл тоо $n > 1$

Boolean төрлийн a массивыг авч үзэх ба элементүүд нь 2 оос n хүртэл дугаарлагдсан бөгөөд бүгд “үнэн” буюу “true” утгатай байна.

for $i = 2, 3, 4, \dots, \sqrt{n}$ – ээс хэтрэхгүй:

if $A[i]$ нь true бол :

for $j = i^2, i^2+i, i^2+2i, i^2+3i, \dots, n$ – ээс хэтрэхгүй:

$A[j] := \text{false}$

Гаралт: $A[i]$ нь true байх бүх i тоог авна (Wikipedia, 2016).

Дээрх хийсвэр алгоритмыг хэрэгжүүлэх чиглэлээр программчлалын C++ хэл дээр бичигдсэн программын нэгэн кодыг авч үзье. Уг кодонд true утгыг 1, false утгыг 0 гэж авч үзсэн байна.

Код 3. Программчлалын C++ хэл, Нэг бүрчлэн шалгасан

#include<iostream>

int $a[10000]$; //10000 элементтэй массив, n нь 10000 хүртэл байж болно

int i, n, j ; // шаардлагатай хувьсагчид

void main()

{ cout << “n=”;

cin>>n; // n - тоог оруулах

for ($i=1; i \leq n; i++$) // бүх өгөгдсөн тоогоор массив үүсгэх, элементийн дугаар нь

$a[i]=1$; // сонгогдох тоог төлөөлөх ба 1 утга оноосноор анхны тоо гэж үзнэ

for ($j=2; j \leq n/2; j++$) // бүх хуваагчдыг сонгох

for ($i=2; i \leq n; i++$) // олонлогийн бүх тоог түүн авч шалгах

if ($i \% j == 0 \ \&\& \ i != j$) $a[i]=0$; // ээлжит тоог шалгах ба зохиомол бол 0 утга

// оноосноор олонлогоос хасах үйлдлийг гүйцэтгэх

for ($i=1; i \leq n; i++$) if ($a[i]==1$) // массивт үлдсэн анхны тоонуудыг хэвлэх

cout << “ ”; // дугаар нь ээлжит тоог төлөөлнө

Энэхүү алгоритмын дутагдалтай тал нь өгөгдсөн тоонуудаас ямар нэгэн хуваагчид хуваагдах тоонуудыг хасах үедээ сонгогдсон хуваагчид хуваагдахгүй ч, өмнөх алхамуудад өөр хуваагчуудад хуваагдаж анхны олонлогоос хасагдсан тоонуудыг ч нэг бүрчлэн шалгаж байгаа явдал бөгөөд уг программыг сайжруулахын тулд шалгах нөхцлийг дараах байдлаар бичиж болно.

if ($a[i]==1 \ \&\& i \% j == 0 \ \&\& \ i != j$) $a[i]=0$;

Гэвч энэ нь төгс төгөлдөр шийдэл биш юм. Учир нь нөхцлийг хичнээн сайжруулсан ч дээр дурьдсан хийсвэр алгоритмд гарсан анхны тоон i хуваагчийн хувьд $i^2, i^2+i, i^2+2i, i^2+3i, \dots$ хэлбэртэй n – ээс хэтрэхгүй тоонууд зохиомол тоо байна гэсэн тодорхой үр дүнг ашиглахгүй байгаа явдал юм.

Иймд энэ үр дүнг ашиглах чиглэлээр программын ажиллах хугацааг багасгах нэгэн хувилбар программын кодыг сонирхоё. Үүнд i тоо анхны тоо бол $2i, 3i, \dots$ хэлбэртэй n – ээс хэтрэхгүй тоонууд зохиомол тоо байна гэсэн үндэслэлээр өгөгдсөн тоон олонлогоос хасах зорилго тавилаа.

Код 4. Программчлалын C++ хэл, Давтамж ашигласан

#include<iostream>

#include<math.h>

int n, j, k, b ;

int $ok[30000]$; //хувьсагчдыг зарлах, ok массив нь бүх тоон олонлогийг төлөөлнө

void main()

{ cout << “n(<30000)=”; cin>>n; // өгөгдөл оруулах

$ok[1]=0$; //1- ийг анхны тоо биш гэж үзэх

for ($j=2; j \leq n; j++$) // 2 - оос n хүртэлх бүх тоонуудыг авах

$ok[j]=1$;

$b=\text{sqrt}(n)$; // хуваагчуудын хамгийн их хуваагч

$k=0$;

while ($k \leq b$) // анхны тоо мөн эсэхийг шалгаж зохиомол тоог хасах

{while (! $ok[k]$) // ээлжит хамгийн бага анхны тоог олох

$k++$;

$j=k*2$; // ээлжит хамгийн бага тоонд хуваагдах хамгийн бага зохиомол тоо үүсгэх

while ($j \leq n$)

{ //зохиомол тоонуудыг анхны тоон k алхамтайгаар массиваас хасах

$ok[j]=0$;

$j=j+k$;

//энэ давталт нь анхны олонлогийн тоонуудыг

```

нэг бүрчлэн шалгахаас сэргийлнэ
} //end while
} //end while
for (j=1; j<= n; j++) // олонлогт үлдсэн тоонууд
буюу анхны тоонуудыг хэвлэх
if (ok [ j ]) cout<< j<<" "; //массивын элементийн
дугаар нь ээлжит тоог илтгэнэ
} //end main

```

Дүгнэлт

Дэвшүүлсэн бодлогыг бодох алгоритмыг массив ашиглан шийдсэнээр уг алгоритмыг зөвхөн олонлог төрлийн хувьсагчийн хувьд ашигладаг гэсэн өрөөсгөл ойлголтыг үгүйсгэж, илүү олон элемент агуулах өгөгдлийн нийлмэл төрлүүдийн тусламжтай программчлах нь алгоритмын үнэт чанарыг илүү тодорхой гаргаж өгнө.

Тавигдсан бодлогыг массив ашиглан шийдсэнээр 255 хүртэлх тоонд хязгаарлагдах биш массивын элементийн тоотой тэнцүү өөрөөр хэлбэл 30000 хүртэлх тоонуудаас анхны тоонуудыг нь уг алгоритмын тусламжтай олох боломжийг бүрдүүлж байна.

Аливаа программчлал нь хязгаарлагдмал нөөц (санах ой гэх мэт) дээр хийгддэг тул цаашид өгөгдлийн динамик бүтцийг ашиглан шийдвэл илүү үр дүн үзүүлэх боломжтой юм.

POLICY ANALYSIS AND SOLUTIONS TO FIND SEQUENTIAL PRIME NUMBERS.

Abstract

In the framework of Informatics and Algorithm programming lesson which reflected on the curriculum of secondary school and university, prime related problems and algorithms contain the dominant percent of total problems and algorithms. In this way, prime related problems and algorithms are used as a basis during the process of founding multi-task programming. Therefore, this article considers the prime related one problem and its solution of The Sieve of Eratosthenes algorithm. Furthermore, the aim was set for improving and revising one code written by the Pascal programming based upon document facts and analyzing process.

Зохиогчийн тухай

Дамдинсүрэнгийн Бүрэн-Арвижих, МУИС – ийн Ховд дахь салбар сургуулийг 2001 онд математик мэдээлэл зүйн багш мэргэжлээр төгссөн, Боловсрол судлалын магистр зэрэгтэй. Ховд их сургуулийн Байгалийн шинжлэл, технологийн сургуулийн Математик, мэдээлэл зүйн тэнхимд 20 дахь жилдээ багшаар ажиллаж байна. Боловсролд мэдээлэл харилцааны технологийг нэвтрүүлэх арга зүйн чиглэлээр судалгаа хийдэг

Ашигласан ном

1. Гармаа.Д. (2009). Алгоритмчлал ба программчлал. Улаанбаатар.
2. Чойжоованчиг.Л, Цэдэвсүрэн.Д, & Хатанхангай.Ц. (2005). Мэдээлэл зүйн практикум. Улаанбаатар.
3. Wikipedia. (2016). Retrieved from Sieve of Eratosthenes: https://en.wikipedia.org/wiki/Sieve_of_Eratosthenes#cite_note-horsley-1