



Ж.Дашдэмбэрэл
МУБИС-ийн МБУС-ийн Мэдээлэл зүйн
тэнхимийн багш, магистр



Д.Бүрэн-Арвижix
ХИС-ийн БШТС-ийн МХТ-ийн тэнхимийн
багш, магистр

ШУГАМАН ХАЙЛТЫН АЛГОРИТМУУДЫГ ПАРАМЕТРТ ДАВТАЛТААР ЗОХИОН БАЙГУУЛАХ

Abstract: In computer science, linear search or sequential search is a method for finding a particular value in a list, that consists of checking every one of its elements, one at a time and in sequence, until the desired one is found. Linear search is the simplest search algorithm; it is a special case of brute-force search. Its worst case cost is proportional to the number of elements in the list; and so is its expected cost, if all list elements are equally likely to be searched for. Therefore, if the list has more than a few elements, other methods (such as binary search) will be faster, but they also impose additional requirements.

Key words: search algorithm, linear search, binary search

Амьдрал, практикт болон мэдээлэл зүйн төрөл бүрийн бодлогыг бодох явцад өгөгдлүүдийн ямар нэгэн олонлог, системээс өгөгдсөн элемент, шаардлага хангах өгөгдөл хайж олох асуудал олонтоо тохиолддог. Энэ асуудлыг хэрхэн шийдэх вэ гэдэг нь мэдээлэл зүйн бодлого бодох аргад суралцагчдын зайлшгүй эзэмших мэдлэгийн нэг үндсэн ойлголт бөгөөд ийм төрлийн бодлогыг хайлтын бодлого гэдэг. Энэхүү асуудлыг орчин үеийн программчлалын систем, хөгжүүлэлтийн хүчирхэг орчнууд тусгай зориулсан арга, техник, технологиудыг ашиглан хялбархан шийддэг хэдий ч мэдээлэл зүйн бодлого бодох арга, программчлалын системийг судлах явцдаа хайлтын арга, алгоритмуудыг нарийн сайн ойлгох, уг ойлголтоор дамжуулан программчлалын үндсэн ухагдахуунуудыг таньж мэдэх нь чухал юм.

Мэдээлэлзүйн бодлогыг программчлах үед зайлшгүй эзэмшсэн байх чадвар нь программчлалын хэлний операторуудын дотоод зүй мөн чанар, хүчин чадлыг бүрэн төгс нээн илрүүлж, тэр хэмжээгээрээ бодлого, програмдаа ашиглан өвөрмөц шийдэл, бодолтыг гаргах шаардлагатай байдаг тул шугаман хайлтын арга алгоритмуудыг судлан тэдгээрийн хоорондын ялгаа, онцлог, дотоод мөн чанарыг нь нээн илрүүлэх явцдаа шугаман хайлтын алгоритмуудыг программчлалын Си хэлний параметрт давталтын тусламжтайгаар зохион байгуулах зорилго дэвшүүлээ.

Дэвшүүлсэн зорилгын хүрээнд

- Шугаман хайлтын алгоритмуудыг судлан, тэдгээрийн онцлогийг гаргах
- Шугаман хайлтын алгоритмуудыг параметрт давталт ашиглан шийдэх

Өгөгдлүүдийн ямар нэгэн систем дотор өгөгдсөн элемент байна уу, үгүй юу, байгаа бол чухам хаана байна вэ? гэсэн асуудлыг авч үзэх бодлогыг хайлтын бодлого гэх ба энэ бодлого нь ихэвчлэн массивын элементүүд дотроос өгөгдсөн утга бүхий элементийг хайх бодлогод шилждэг. Өгөгдсөн массиваас өгсөн элементийг хайх явцад уг массивын талаар ямар нэгэн мэдээлэл өгөгдөөгүй тохиолдолд хайлтыг зөвхөн шугаман хэлбэрийн хайлтаар явуулахаас өөр аргагүй байдаг. Энд

хийгдэх жишилтийн тоогоороо ялгагдах хоёр алгоритм байдаг тул товч байдлаар авч үзье.

Бодлого: Өгсөн n ширхэг элементтэй a – шугаман массиваас x – элементийг хайж ол.

1. **Шугаман хайлт:** Энэ алгоритмын үндсэн санаа нь массивын элемент бүрийг эхний элементээс эхлэн (сүүлийн элементээс эхлэн) нэг бүрчлэн сонгож, сонгогдсон элементийн утгыг өгөгдсөн элементтэй жишин хэрэв тэдгээрийн утгууд давхцаж байвал массивын тухайн элементийн дугаарыг нэмэлт хувьсагчид хадгалах явдал юм. Эцэст нь нэмэлт хувьсагчийн утгыг шалгана. [3.89]

Хувилбар 1.1:

```
for (i=1; i<=n; i++) if (a[i]==x) k=i; /* эхний элементээс эхлэн сонгох, хайж буй элемент олдвол түүний дугаарыг хадгалах */ if (k<=n) //үр дүн хэвлэх printf ("Хайж буй элемент %d –р байранд байна",k); else printf("Хайсан элемент олдсонгүй"); [2.34]
```

Энэ аргын гол дутагдал нь

- хайж буй элементийг олсон хэдий ч цаашид массивын элементүүдийг шалгаж дуустал хайлтыг үргэлжлүүлдэг
- хайж буй элемент хэд хэд орсон бол сүүлийн (эхний) элементийн дугаарыг олдогт оршдог тул эдгээр дутагдлыг засах өөрөөр хэлбэл хайж буй элемент олдмогц хайлтыг зогсоохын тулд хайлтыг дараах байдлаар зохион байгуулдаг.

Хувилбар 1.2:

```
i=1; //массивын эхний элементийг сонгох while (i<=n && a[i]!=x)/*массивын элемент шалгагдаж дуусаагүй, хайж буй элемент олдоогүй үед хайлтыг үргэлжлүүлэх давталт */i++;/*хайж буй элемент олдоогүй үед дараагийн элементийг сонгохын тулд индексийг өөрчлөх*/if (i<=n) //үр дүн хэвлэх printf("Хайж буй элемент %d –р байранд байна",i); else printf("Хайсан элемент олдсонгүй"); [1.327]
```

2. **Саад бүхий шугаман хайлт:** Энэ алгоритмын үндсэн санаа нь массивын $0 - p$ ($n+1 - p$) байранд хайж буй x – элементийг байрлуулж түүнийг "саад" гэж нэрлэдэг. Үүний гол зорилго нь дээрх алгоритмын биелэлтийг удаашруулж буй давталтын толгойд бичигдсэн давхар нөхцөлийг хялбарчлахад оршдог. Өөрөөр хэлбэл, хайлт явуулж буй a – массивд хайж байгаа x – элементийг $0 - p$ ($n+1 - p$) байранд байрлуулснаар хайлтыг массивын төгсгөлөөс (эхнээс) эхлэн явуулахдаа давталтын нөхцөлд зөвхөн ($a[i] != x$) нөхцөлийг үлдээх боломжтой болно. Энэ нөхцөл хайлтыг явуулах үед хайж буй x – элемент өгөгдсөн a – массивд байхгүй тохиолдолд i – индексийн $0 (n+1)$ утганд эсрэг утгаа аван давталт дуусна. Давталтыг дуусгах нөхцөл нь "саад" гэж нэрлэгдэх элемент болсон тул массивын элементүүдийг бүгдийг нь шалгаж дууссан эсэхийг хянах шаардлагагүй байна. Давталт хийсний дараа i – индексийн утгыг "саад" элементийн индексийн утгаас (0 эсвэл $n+1$) ялгаатай эсэхийг шалган үр дүнг хэвлэнэ. [1.90]

Хувилбар 2:

```
a[0]=x;
/*0 – p байранд x – элементийг байрлуулж "саад" үүсгэх*/
i=n; /*"саад" элемент массивын эхэнд байрласан учир хайлтыг массивын төгсгөлөөс эхэн рүү чиглэн явуулахын тулд төгсгөлийн элементийн индекс авах */
while (a[i] != x)
// хайлтыг явуулах давталт
i --;
/*хайж буй элемент олдоогүй бол өмнөх элементийг шалгахын тулд индексийг өөрчлөх*/
if (i != 0) //үр дүн хэвлэх
printf("Хайж буй элемент %d –р байранд байна", i);
else printf("Хайсан элемент олдсонгүй");
```

Эндээс энэ хоёр алгоритмын шинжүүдийг дараах байдлаар гаргаж болно.

Шугаман хайлт:

1. Ямар ч массивын хувьд ашиглаж болно
2. Давтах нөхцөл нь НИЙЛМЭЛ нөхцөл байна
3. Давталтын биелэлт УДААН
4. Ямар ч нэмэлт боловсруулалт ХИЙХГҮЙ

Саад бүхий шугаман хайлт:

1. Ямар ч массивын хувьд ашиглаж болно
2. Давтах нөхцөл нь ЭНГИЙН нөхцөл байна



3. Давталтын биелэлт ХУРДАН

4. “Саад” элемент байрлуулахын тулд нэмэлт боловсруулалт хийнэ.

Авч үзсэн хоёр аргын хувьд зарчмын хувьд хэд хэдэн ялгаа байгаа нь харагдаж байна. Мөн хайлтын процессдоо эхэндээ нөхцөлтэй давталтыг ашиглаж байгаа хэдий ч аль ч нөхцөлт давталтын хувьд дээрх ялгаа нь адилхан оршсоор байх болно. Харин эдгээр сайжруулсан алгоритмыг параметрт давталтаар хэрхэн шийдэх нь ихээхэн сонирхолтой юм.

Программчлалын Си хэлний параметр давталт нь өөртөө эхэндээ нөхцөлтэй давталтын шинжийг бүрэн агуулсан байдаг тул [4] эдгээр алгоритмыг параметрт давталтын тусламжтайгаар шийдэх бүрэн боломжтой.

Параметрт давталт ашиглан шугаман хайлт гүйцэтгэх алгоритмын хийсвэр код нь дараах хэлбэртэй бичигдэнэ.

Хийсвэр код:

```
Function LinearSearch(L[0:n-1],x)
Input: L[0:n-1], x
Output: x элементийн массив дахь байрлал дахь дугаар эсвэл -1
for i 0 to n - 1 do
if x = L[i] then return (i)
endif
endfor
return (-1)
end LinerSearch
```

Энэхүү хийсвэр код дээр тулгуурлан боловсруулалтыг гүйцэтгэе.

Хувилбар 1.1 кодыг параметрт давталт ашиглан шийдсэн тул түүнийг сайжруулахын тулд дараах байдлаар бичиж болно. Үүний тулд нэгэн операторыг авч үзье.

- break оператор :

Программд break оператор нь нөхцөл шалгах, сонголтын болон давталтын операторуудын дотор буюу хэрэгжүүлэх операторын бие дотор хэрэглэгддэг бөгөөд гүйцэтгэх үүрэг нь эдгээр операторын биелэлтийг тасалж программын удирдлагыг дараагийн операторт шилжүүлэхэд оршино.

Хувилбар 3:

```
for (i=1; i<=n; i++) if (a[ i ]==x) break; /* эхний элементээс эхлэн сонгох, хайж буй элемент олдмогц давталтыг таслах*/
if (i<=n) //үр дүн хэвлэх
printf("Хайж буй элемент %d –р байранд байна",i);
else printf("Хайсан элемент олдсонгүй");
```

Энд break оператор ашигласнаар Хувилбар 1.1 –ийн дээр дурдсан хоёр дутагдал арилахын зэрэгцээ нэмэлт хувьсагч ашиглахгүй байгаа нь бас нэг давуу тал болно.

Харин Хувилбар 1.2 – ийг параметрт давталтаар шууд бичиж болно.

Хувилбар 4:

```
i=1;//массивын эхний элементийг сонгох
for ( ;i<=n && a[i]!=x; ) /*массивын элемент шалгагдаж дуусаагүй, хайж буй элемент олдоогүй үед хайлтыг үргэлжлүүлэх давталт */
i++;/*хайж буй элемент олдоогүй үед дараагийн элементийг сонгохын тулд индексийг өөрчлөх*/
if (i<=n) //үр дүн хэвлэх
printf("Хайж буй элемент %d –р байранд байна",i);
else printf("Хайсан элемент олдсонгүй");
Энэ нь параметрт давталтын шинж чанарыг ашиглан бичиж байгаа боловч үнэн хэрэгтээ
```

Хувилбар 3 – д авч үзсэн if (a[i]==x) break; операторын нөхцөлийг уг параметрт давталтын нөхцөлд нэгтгэн оруулж байгаа хэрэг юм.

Хувилбар 4 –ийн давталтын хэсгийг дараах байдлаар бичиж болох ба гол анхаарах зүйл нь параметрт давталт нь өөрийн ард бичигдсэн зөвхөн нэг операторыг давтан биелүүлдэг тул энэ давтан биелүүлэх оператор нь зайлшгүй хоосон оператор байх ёстой.

- Хоосон оператор: Ямар ч үйлдэл боловсруулалт хийхгүй бөгөөд зөвхөн цэгтэй таслал (;) –аас тогтоно.

Хувилбар 5:

```
for ( i=1;i<=n && a[i]!=x; i++)
/*эхний элементээс эхлэн, массивын бүх элемент шалгагдаж дуусаагүй, хайж буй элемент олдоогүй
үед хайлтыг үргэлжлүүлэх давталт */
; //хоосон оператор
if (i<=n) //үр дүн хэвлэх
printf("Хайж буй элемент %d –р байранд байна",i);
else printf("Хайсан элемент олдсонгүй");
```

Нэгэнт параметр давталт нь хоосон оператор биелүүлэхийн зэрэгцээ, давтах нөхцөлөөсөө шалтгаалан давталтыг дуусгаж чаддаг тул Хувилбар 2 буюу саад бүхий шугаман хайлтыг дараах байдлаар бичиж болно.

Хувилбар 6:

```
a[0]=x; /*0 – р байранд x – элементийг байрлуулж "саад" үүсгэх*/
i=n; /*"саад" элемент массивын эхэнд байрласан учир хайлтыг массивын төгсгөлөөс эхэн уруу
чиглэн явуулахын тулд төгсгөлийн элементийн индекс авах */
for(;a[i] != x; )
// хайлтыг явуулах давталт
i --; /*хайж буй элемент олдоогүй бол өмнөх элементийг шалгахын тулд индексийг өөрчлөх*/
if (i != 0) //үр дүн хэвлэх
printf("Хайж буй элемент %d –р байранд байна", i);
else printf("Хайсан элемент олдсонгүй");
```

Эхний элементийн сонголт, индексийн өөрчлөлтийг давталт дотор оруулан бичвэл энэ алгоритм нь

```
a[0]=x; for (i=n; a[i]!=x; i--);
if (i != 0)
printf("Хайж буй элемент %d –р байранд байна", i);
else printf("Хайсан элемент олдсонгүй");
```

хэлбэртэй бичигдэнэ.
Энэ нь бусад бичиглэлтэй харьцуулахад хамгийн товч ойлгомжтой бичиглэл болно.

ДҮГНЭЛТ

Шугаман хайлтын алгоритмуудыг судалснаар тэдгээрийн нарийн ялгааг гаргахын сацуу эдгээр алгоритмыг зарим ойлголт, операторуудыг ашиглан параметрт давталтаар бүрэн гүйцэд шийдэж болох ба нэг бодлогыг олон хувилбар, арга замаар программчлах нь программчлалын үндсэн операторуудыг төгс эзэмших, бодлогын тавил, түүнийг бодох алгоритмыг гүн гүнзгий ойлгох боломжийг бүрдүүлдэг гэж үзлээ.

АШИГЛАСАН МАТЕРИАЛ

Цэдэвсүрэн, Д., Хатанхангай, Ц., Чойжоованчиг, Л. (1999). *Мэдээлэл зүйн практикум Turbo Pascal 7.0*. Улаанбаатар

Kenneth A. Berman., Jerome L. Paul. (2005). *ALGORITHMS sequential, parallel and distributed*, USA

Могилев А.В., Пак Н.И., Хеннер, Е.К. (2000). "Информатика" под редакцией Хеннера Е.К., М.- Академия Минжирмаа, Р. (2011). Параметрт давталтын харьцуулалт. *Баруун монгол, түүний хил залгаа нутгийн ард түмний соёл, байгалийн нөөц*. Олон улсын X хурал. Ховд