

Алгоритмыг хийсвэр кодоор дүрслэх

Б.Наранчимэг

Мэдээлэл, компьютерийн ухааны тэнхим



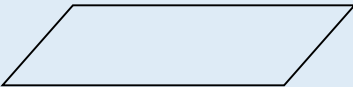

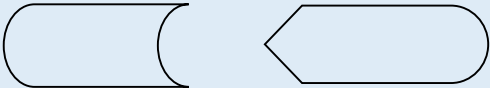
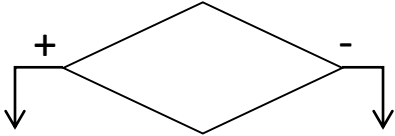
ХШУИС, МУИС

naranchimeg@seas.num.edu.mn

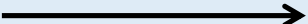

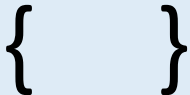
Өмнөх хичээлээр

- Алгоритмын үндсэн алхмууд
- Алгоритмыг блок схемээр дүрслэх

Блок схемийн элементүүд

Элемент	Дүрслэл	Үйлдэл
Эхлэл / Төгсгөл		Алгоритмын эхлэл төгслөл
Хувьсагч зарлах		Алгоритмд хэрэглэх хувьсагч зарлах
Утга оруулах		Хувьсагчид гаднаас өгөгдөл оруулах
Утга олгох		Утга олгох
Хэвлэх		С хувьсагчийн утгыг хэвлэх
Нөхцөл шалгах		Нөхцөл шалгах Салаа замын аль нэгээр нь явна. +, 1, yes, у тухайн нөхцөл биелж байгаа -, 0, no, n тухайн нөхцөл биелээгүй

Блок схемийн элементүүд

Элемент	Дүрслэл	Үйлдэл
Шилжилтийн шугам		Шилжилтийн шугам
Холбоос		Тухайн хуудсанд багтахгүй байгаа блок схемийг өөр хуудаснаас үргэлжлүүлэхдээ хоёр хуудасны холбоос
Тайлбар		Ямар нэгэн алхамын гүйцэтгэх үүргийг дэлгэрэнгүй тайлбарлахад хэрэглэнэ.

Хийсвэр код (Pseudo code)

- Алгоритмыг дүрслэх өргөн хэрэглэгддэг хэлбэр
- Уншиж бичихэд хялбар
- Асуудлын логик дээр анхаарлаа төвлөрүүлэх боломжийг олгоно
- Англи хэл дээр бүтэцлэгдсэн
- Алгоритмыг програмчлалын хэлээр бичих хэлбэртэй төсөөтэй хэлбэрээр бичигдэнэ.
 - C, C++, Pascal гэх мэт

Хийсвэр код (Pseudo code)

- Дугаарлалт
- Операторууд
 - `if-then-else`
 - `for-to-do`
 - `while-do`
 - `repeat-until or do-while`
- Догол мөр (Indentation) чухал үүрэгтэй
 - Тухайн блокын эхлэл төгсгөлийг заана.

Жишээ нь: Хүснэгтийн хамгийн их элементийг олох

```
1 input (n)
2 input (a[1...n])
3 max ← a[1]
4 for i ← 2 to n
5     do if max < a[i]
6         then max ← a[i]
7 output (max)
```

Жишээ нь: Хүснэгтийн хамгийн их элементийг олох

```
1 input (n)
2 input (a[1...n])
3 max ← a[1]
4 for i ← 2 to n
5   do if max < a[i]
6     then max ← a[i]
7 output (max)
```

Догол мөр нь тухайн алхам аль блокод хамаарч байгааг илэрхийлнэ

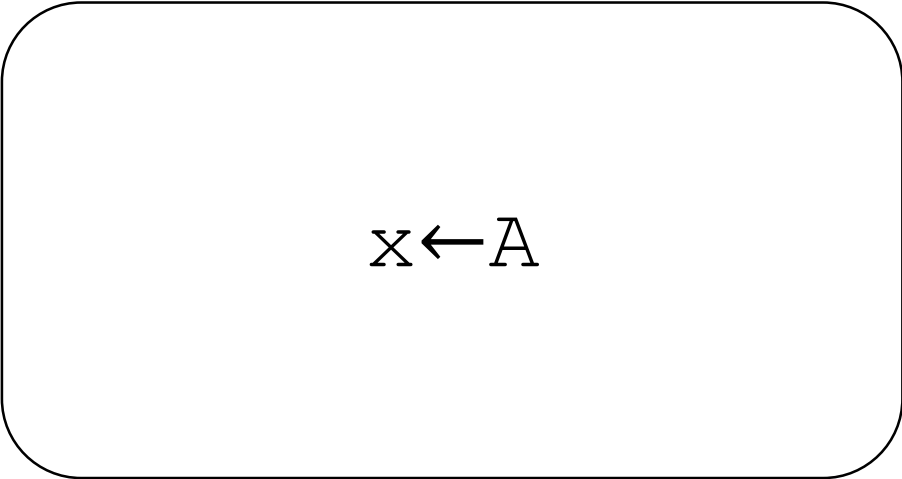
Утга оруулах үйлдэл

- x хувьсагчид утга оруулах
- `input` түлхүүр үг ашиглана.

`input (x)`

Утга олгох үйлдэл

- X нь утга хадгалах хувьсагч
- A нь илэрхийлэл байна.
- \leftarrow ($=$) оператор хэрэглэгдэнэ.



$x \leftarrow A$

Хэвлэж гаргах үйлдэл

- X нь хэвлэж гаргах хувьсагч
- **output** түлхүүр үг ашиглана.

output (x)

Нөхцөл шалгах үйлдэл

- **if-then-else** түлхүүр үгийг хэрэглэнэ.
- **A** нь нөхцөл биелэгдэх алхам
- **B** нь нөхцөл биелэхгүй алхам

```
if НӨХЦӨЛ  
  then A  
  else B
```

```
if sugar is required  
  then add sugar  
  else don't add sugar
```

Нөхцөл шалгах үйлдэл

- **if-then-else** түлхүүр үгийг хэрэглэнэ.
- **A** нь нөхцөл биелэгдэх алхам
- **B** нь нөхцөл биелэхгүй алхам

```
if  НӨХЦӨЛ  
  then A  
  else B
```

Догол мөр!

Нөхцөл шалгах үйлдэл

- **if-then-else** түлхүүр үгийг хэрэглэнэ.
- **A** нь нөхцөл биелэгдэх алхам
- **B** нь нөхцөл биелэхгүй алхам

```
if  НӨХЦӨЛ  
  then A  
  else B
```

Догол мөр!

Нөхцөл шалгах үйлдэл

- **if-then-else** түлхүүр үгийг хэрэглэнэ.
- **A** нь нөхцөл биелэгдэх алхам
- **B** нь нөхцөл биелэхгүй алхам

```
if  НӨХЦӨЛ  
    then A1  
        A2  
    else B1  
        B2
```

Нөхцөл шалгах үйлдэл

- 2 тооны ихийг олох:

```
if A>B  
    then Print A + "is bigger";  
    else Print B + "is bigger";
```


Нөхцөл шалгах үйлдэл

- Зарим хийсвэр кодонд **end** түлхүүр үг хэрэглэнэ

```
if нөхцөл  
  then A  
  else B  
end if
```

```
if A>B  
  then Print A + "is bigger";  
  else Print B + "is bigger";  
end if
```

Давталтыг бичих хэлбэр

- Тоолуурт давталт
 - **for-to-do**
- Нөхцөлт давталт
 - **while-do**
- Давталт нөхцөл
 - **repeat-until**

```
while kettle is not full  
  do keep filling kettle
```

Тоолуурт давталт

- **for-to-do**
- **i**-тоолуур
- **k**- тоолуурын эхний утга
- **m**- тоолуурын төгсгөлийн утга

```
for i ← k to m  
  do A
```

- **A** – давталтын их бие

```
for i ← m downto k  
  do A
```

Тоолуурт давталт

- **n** хүртэлх тоог хэвлэх

1 2 3 4 5 ... n

```
for i ← 1 to n  
  do Print i
```

Нөхцөлт давталт

- **while-do**

```
while НӨХЦӨЛ  
  do A
```

Давталт огт хийгдэхгүй байж болно!

- **A** – давталтын их бие

```
i ← 1  
while i ≤ n  
  do Print i  
      i ← i + 1
```

- **n** хүртэлх тоог хэвлэх

1 2 3 4 5 ... n

Нөхцөлт давталт

```
i ← 1
while i ≤ n
  do Print i
    i ← i + 1
```

- **n** хүртэлх тоог хэвлэх

1 2 3 4 5 ... n

n	Гаралт
5	1 2 3 4 5
1	1
0	?

Давталт огт хийгдэхгүй байж болно!

Нөхцөлт давталт

```
i ← 1
while i ≤ n
  do Print i
    i ← i + 1
```

Давталт огт хийгдэхгүй байж болно!

- **n** хүртэлх тоог хэвлэх

	1	2	3	4	5	...	n
n	Гаралт						
5	1	2	3	4	5		
1	1						
0							

Юу ч хэвлэхгүй!

Давталт нөхцөл

- `repeat-until`

repeat

A

until нөхцөл

- A – давталтын их бие

$i \leftarrow 1$

Repeat

Print i

$i \leftarrow i + 1$

until $i > n$

- n хүртэлх тоог хэвлэх

1 2 3 4 5 ... n

Давталт заавал нэг удаа хийгдэнэ!

Давталт нөхцөл

- **Do-while**

do

A

while нөхцөл

Давталт заавал нэг удаа хийгдэнэ!

- **A** – давталтын их бие

$i \leftarrow 1$

do

Print i

$i \leftarrow i + 1$

while $i \leq n$

- **n** хүртэлх тоог хэвлэх

1 2 3 4 5 ... n

Давталт нөхцөл

```
i ← 1
repeat
  Print i
  i ← i + 1
until i > n
```

- **n** хүртэлх тоог хэвлэх

1 2 3 4 5 ... n

n	Гаралт
5	1 2 3 4 5
1	1
0	?

Давталт дор хаяж 1 заавал хийгдэнэ!

Давталт нөхцөл

```
i ← 1
repeat
  Print i
  i ← i + 1
until i > n
```

- **n** хүртэлх тоог хэвлэх

1 2 3 4 5 ... n

n	Гаралт
5	1 2 3 4 5
1	1
0	1

Давталт дор хаяж 1 заавал хийгдэнэ!

Жишээ бодлого 1: Хүснэгтийн хамгийн их элементийг олох

```
1 input (n)
2 input (a[1...n])
3 max ← a[1]
4 for i ← 2 to n
5     do if max < a[i]
6         then max ← a[i]
7 output (max)
```

```
1 input (n)
2 input (a[1...n])
3 max ← a[1]
4 i ← 2
5 while i ≤ n
6     do if max < a[i]
7         then max ← a[i]
8     i ← i + 1
9 output (max)
```

Жишээ бодлого 2: Эвклидийн алгоритм

```
1. input  (n)
2. input  (m)
3. repeat
4.     while m < n
5.         do n ← n - m
6.     while m > n
7.         do m ← m - n
8. until m == n
9. output (m)
```

Жишээ бодлого 2: Эвклидийн алгоритм

```
1. input  (n)
2. input  (m)
3. while (n!=m)
4.     do if n > m
5.         then n←n-m
6.         else m←m-n
7. output  (m)
```

Програмчлалын хэл Си

1. **input** (n)
2. **input** (m)
3. **while** (n!=m)
4. **do** if n > m
5. **then** n←n-m
6. **else** m←m-n
7. **output** (m)

```
int main()
{
    int n, m;
    printf("n="); scanf("%d", &n);
    printf("m="); scanf("%d", &m);
    while (n != m)
    {
        if (n > m)
        {
            n = n - m;
        }
        else
        {
            m = m - n;
        }
    }
    printf("Hamgiin ih erunhii huvaagch=%d", m);
    return 0;
}
```

goto команд

- Алгоритмын удирдлагыг өөр алхам руу шилжүүлэхдээ ашиглана.
- Аль болох ашиглахгүй байвал дараа нь програмчлалын хэлэнд хөрвүүлэхэд амар байдаг.
 - Ихэнх програмын хэл goto командыг дэмждэггүй!

Асуулт?

Даалгавар

1. Хэрвээ n нь 10-аас эрс их бол k -д 1-ийг, эсрэг тохиолдол k -д -1-ийг олгох үйлдлийг хийсвэр кодоор бич.
2. Дараах давталт хийгдсэний дараа a , i хувьсагчдын утга хэдтэй тэнцүү байх вэ?

```
1. a ← 0
2. for i ← 1 to 10
3.     do a ← a + i
```

2a.

```
1. a ← 1
2. i ← 1
3. while i < 5
4.     do a ← a*i
5.     i ← i+1
```

2b.

```
1. a ← 2
2. i ← 3
3. repeat
4.     a ← a*i
5.     i ← i-1
6. until i==0
```

2c.