

Adatszerkezetek és algoritmusok

Gyakorlat

Hannusch Carolin

DEIK

2026

Az algoritmus fogalma

- ▶ Egy algoritmus egy számítógépes eljárás, mely egy értékből, vagy értékek egy halmazából kiindulva gyárt egy (másik) értéket, vagy értékek halmazát. (Input - Output)

Az algoritmus fogalma

- ▶ Egy algoritmus egy számítógépes eljárás, mely egy értékből, vagy értékek egy halmazából kiindulva gyárt egy (másik) értéket, vagy értékek halmazát. (Input - Output)
- ▶ Egy algoritmus egy eszköz egy számítási probléma megoldására.

Az algoritmus fogalma








- ▶ Egy algoritmus egy számítógépes eljárás, mely egy értékből, vagy értékek egy halmazából kiindulva gyárt egy (másik) értéket, vagy értékek halmazát. (Input - Output)
- ▶ Egy algoritmus egy eszköz egy számítási probléma megoldására.
- ▶ Az input-output kapcsolat problémafüggő.

Folyamatábra

Building blocks [\[edit \]](#)

Common symbols [\[edit \]](#)

The [American National Standards Institute](#) (ANSI) set standards for flowcharts and their symbols in the 1960s.^[14] The [International Organization for Standardization](#) (ISO) adopted. Generally, flowcharts flow from top to bottom and left to right.^[17]

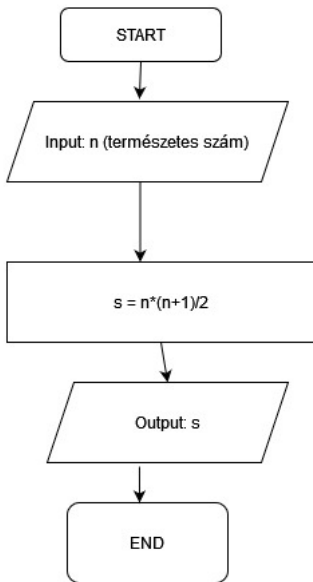
ANSI/ISO Shape	Name	Description
	Flowline (Arrowhead) ^[15]	Shows the process's order of operation. A line coming from one symbol and pointing at another. ^[14] Arrowheads are added if the flowchart is a process, such as "submit inquiry" or "receive product".
	Terminal ^[14]	Indicates the beginning and ending of a program or sub-process. Represented as a stadium , ^[14] oval or rounded (fillet) rectangle. T
	Process ^[15]	Represents a set of operations that changes value, form, or location of data. Represented as a rectangle . ^[15]
	Decision ^[15]	Shows a conditional operation that determines which one of the two paths the program will take. ^[14] The operation is commonly a y
	Input/Output ^[15]	Indicates the process of inputting and outputting data, ^[15] as in entering data or displaying results. Represented as a parallelogram .
	Annotation ^[14] (Comment) ^[15]	indicating additional information about a step in the program. Represented as an open rectangle with a dashed or solid line connect
	Predefined Process ^[14]	Shows named process which is defined elsewhere. Represented as a rectangle with double-struck vertical edges. ^[14]
	On-page Connector ^[14]	Pairs of labeled connectors replace long or confusing lines on a flowchart page. Represented by a small circle with a letter inside. ^[14]
	Off-page Connector ^[14]	A labeled connector for use when the target is on another page. Represented as a home plate-shaped pentagon . ^[14] ^[18]

A tea főzésének folyamatábrája

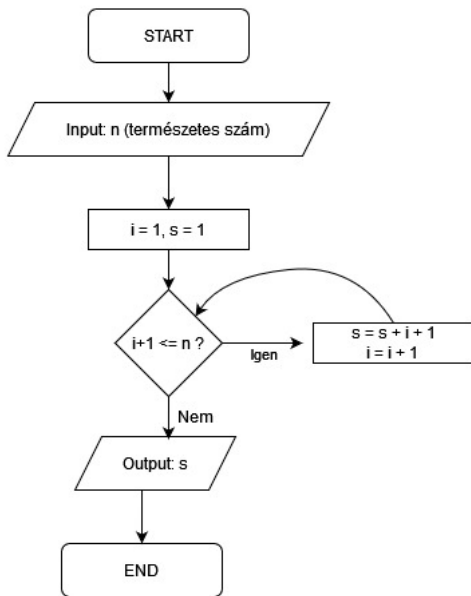


Adjuk össze a számokat 1-től n-ig

Adjuk össze a számokat 1-től n-ig



...és n lépéssel



```
>>> def osszeadrek(n):  
...     x = 0  
...     i = 1  
...     while i < n+1:  
...         x = x + i  
...         i = i + 1  
...     return x  
...  
>>> osszeadrek(10)  
55
```

```
>>> def osszeadas(n):  
...     sum = n*(n+1)/2  
...     return sum  
...  
>>> osszeadas(10)  
55.0  
>>> osszeadas(9)  
45.0
```

```
>>> osszeadas(10)
55.0
>>> def osszeadrek(n):
...     x=0
...     i=1
...     while i<n+1:
...         x=x+i
...         i=i+1
...
>>> def osszeadrek(n):
...     x=0
...     i=1
...     while i<n+1:
...         x=x+i
...         i=i+1
...         return x
...
>>> osszeadrek(10)
1
>>> def osszeadrek(n):
...     x=0
...     i=1
...     while i<n+1:
...         x=x+i
...         i=i+1
...     return x
...
>>> osszeadrek(10)
55
```

```
0.0001583099365234375
```

```
>>> import time
```

```
...
```

```
... # Save timestamp
```

```
... start = time.time()
```

```
...
```

```
... print(osszead(1000))
```

```
...
```

```
... # Save timestamp
```

```
... end = time.time()
```

```
...
```

```
... print(end - start)
```

```
...
```

```
500500.0
```

```
9.322166442871094e-05
```

```
>>> import time
```

```
...
```

```
... # Save timestamp
```

```
... start = time.time()
```

```
...
```

```
... print(osszeadrek(1000))
```

```
...
```

```
... # Save timestamp
```

```
... end = time.time()
```

```
...
```

```
... print(end - start)
```

```
...
```

```
500500
```

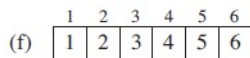
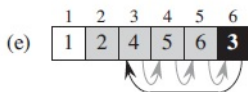
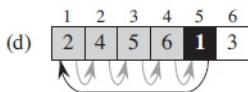
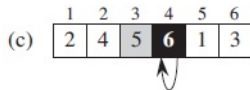
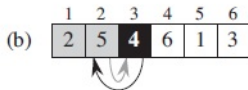
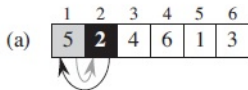
```
0.00016546249389648438
```

```
>>> |
```

Algoritmus rendezésre

Input: $A = \langle 5, 2, 4, 6, 1, 3 \rangle$

Output: $A' = \langle 1, 2, 3, 4, 5, 6 \rangle$



Pszeudokód

```
for  $j = 2$  to  $A.length$  do
   $key = A[j]$ 
   $i = j - 1$ 
  while  $i > 0$  do
    if  $A[i] > key$  then
       $A[i + 1] = A[i]$ 
       $i = i - 1$ 
       $A[i + 1] = key$ 
    end if
  end while
end for
```

▷ $A[j]$ elhelyezése $A[1..j-1]$ -ben.

Lineáris keresés

Input: egy vektor

Lineáris keresés

Input: egy vektor

Output: a vektorban megadott érték helye

Lineáris keresés

Input: egy vektor

Output: a vektorban megadott érték helye

Az algoritmus indul a vektor első helyénél, és így halad előre.

Mindegyik értéknél megvizsgálja, hogy az adott érték megegyezik-e a keresett értékkel.

Lineáris keresés

Input: egy vektor

Output: a vektorban megadott érték helye

Az algoritmus indul a vektor első helyénél, és így halad előre. Mindegyik értéknél megvizsgálja, hogy az adott érték megegyezik-e a keresett értékkel. Ha a vektor hossza n -gyel, akkor az algoritmus lépés száma n -gyel legfeljebb. Az algoritmus futási ideje legfeljebb n .

Bináris keresés

Input: egy rendezett vektor

Bináris keresés

Input: egy rendezett vektor

Output: a vektorban megadott érték helye

Bináris keresés

Input: egy rendezett vektor

Output: a vektorban megadott érték helye

Az algoritmus ketté vágja a megadott vektort, a közepén talált érték vagy kisebb vagy nagyobb, mint a keresett érték. Emiatt le lehet vágni az egyik felét a vektornak.

Bináris keresés

Input: egy rendezett vektor

Output: a vektorban megadott érték helye

Az algoritmus ketté vágja a megadott vektort, a közepén talált érték vagy kisebb vagy nagyobb, mint a keresett érték. Emiatt le lehet vágni az egyik felét a vektornak. Az algoritmus futási ideje legfeljebb $\log_2 n + 1$

```
def linear_search(A, x, n):  
    for i in range(n):  
        if A[i] == x:  
            return i  
    return unsuccessful
```

```
print(linear_search([1, 2, 3, 4, 5], 3, 5))
```

```
> def binary_search(A, n, x):  
    L = 0  
    R = n-1  
    while L <= R:  
        m = math.floor((L+R)/2)  
        if A[m] < x:  
            L = m + 1  
        elif A[m] > x:  
            R = m - 1  
        else:  
            return m  
    return unsuccessful
```

```
...  
>>>  
>>> binary_search([1,2,3,4,5],5,2)  
1  
>>> binary_search([1,2,3,4,5],5,1)  
0  
>>> binary_search([1,3,4,5,7,9,33],7,9)  
5
```

Feladatok I

1. Tervezen algoritmust egy hétköznapi folyamathoz (pl. ebédfőzés, táska bepakolás). Adja meg a folyamatábrát és pszeudokódot!
2. Adott egy egész számokat tartalmazó lista és egy k egész szám. Írj algoritmust, amely lineáris kereséssel meghatározza, hogy k szerepel-e a listában. Ha igen, add meg az indexét, ellenkező esetben -1 legyen az eredmény.
3. Adott egy lista és egy k szám. Határozd meg lineáris kereséssel az **összes** olyan indexet, ahol k előfordul a listában.
4. Adott egy egész számokat tartalmazó lista. Lineáris kereséssel keresd meg a lista **első páros számát**. Ha nincs ilyen elem, írd ki, hogy „Nincs páros szám”.

Feladatok II

1. Adott egy nem üres lista. Határozd meg lineáris kereséssel a lista **legnagyobb elemét** és annak indexét.
2. Adott egy rendezett lista és egy k szám. Határozd meg bináris kereséssel, hogy k szerepel-e a listában, és ha igen, add meg az indexét.
3. Adott egy rendezett lista és egy k szám, amely nem szerepel a listában. Határozd meg azt az indexet, ahová k beszúrható úgy, hogy a lista rendezett maradjon.
4. Adott egy rendezett lista, amelyben egy elem többször is előfordulhat, valamint egy k szám. Határozd meg bináris kereséssel k **első előfordulásának** indexét.
5. Magyarázd meg röviden, miért nem alkalmazható a bináris keresés rendezetlen listára.