Adatszerkezetek és algoritmusok

Gyakorlat

Hannusch Carolin

DEIK

2023. április 15.

Félév beosztás

- 1. dolgozat 6. gyakorlati órán

Félév beosztás

- 1. dolgozat 6. gyakorlati órán
- 2. dolgozat 12. gyakorlati órán

Félév beosztás

- 1. dolgozat 6. gyakorlati órán
- 2. dolgozat 12. gyakorlati órán
- javító/pót dolgozat 13. gyakorlati órán

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.

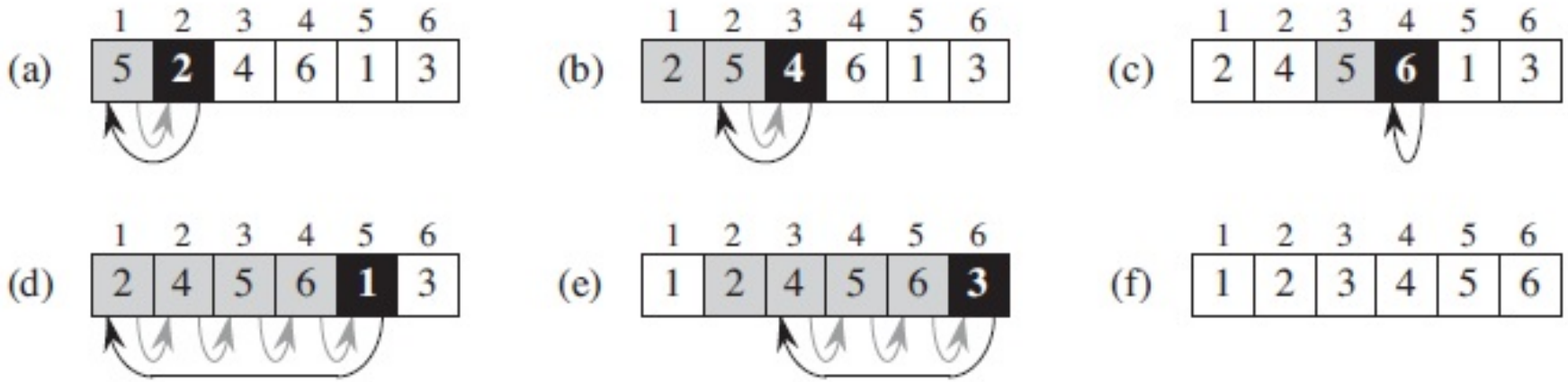
Az algoritmus fogalma

- 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.

Algoritmus rendezésre

Input: A=<5,2,4,6,1,3>

Output: A'=<1,2,3,4,5,6>



Pszeudokód

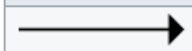
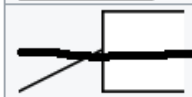
for $j = 2$ to $A.length$ **do**
    $key = A[j]$ ▷ A[j] elhelyezése A[1..j-1]-ben. State $i = j - 1$
    while $i > 0$ **do**
        $A[i] > key$
        $A[i + 1] = A[i]$
        $i = i - 1$
        $A[i + 1] = key$
    end while
end for

Folyamatábra

Building blocks [edit]

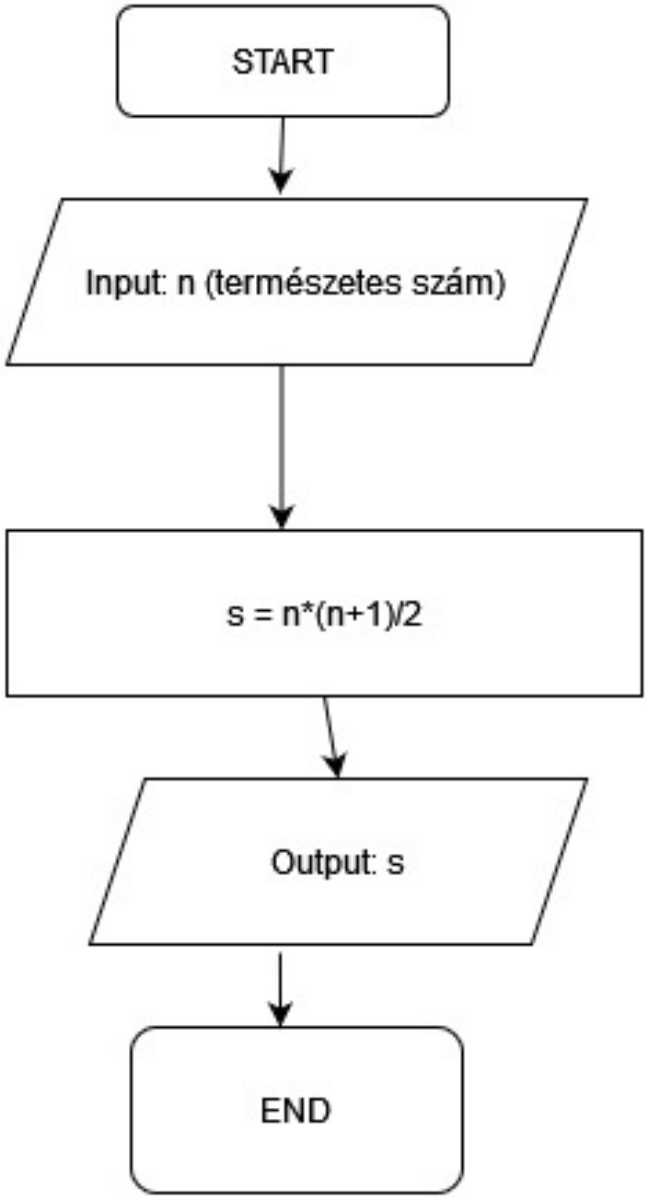
Common symbols [edit]

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

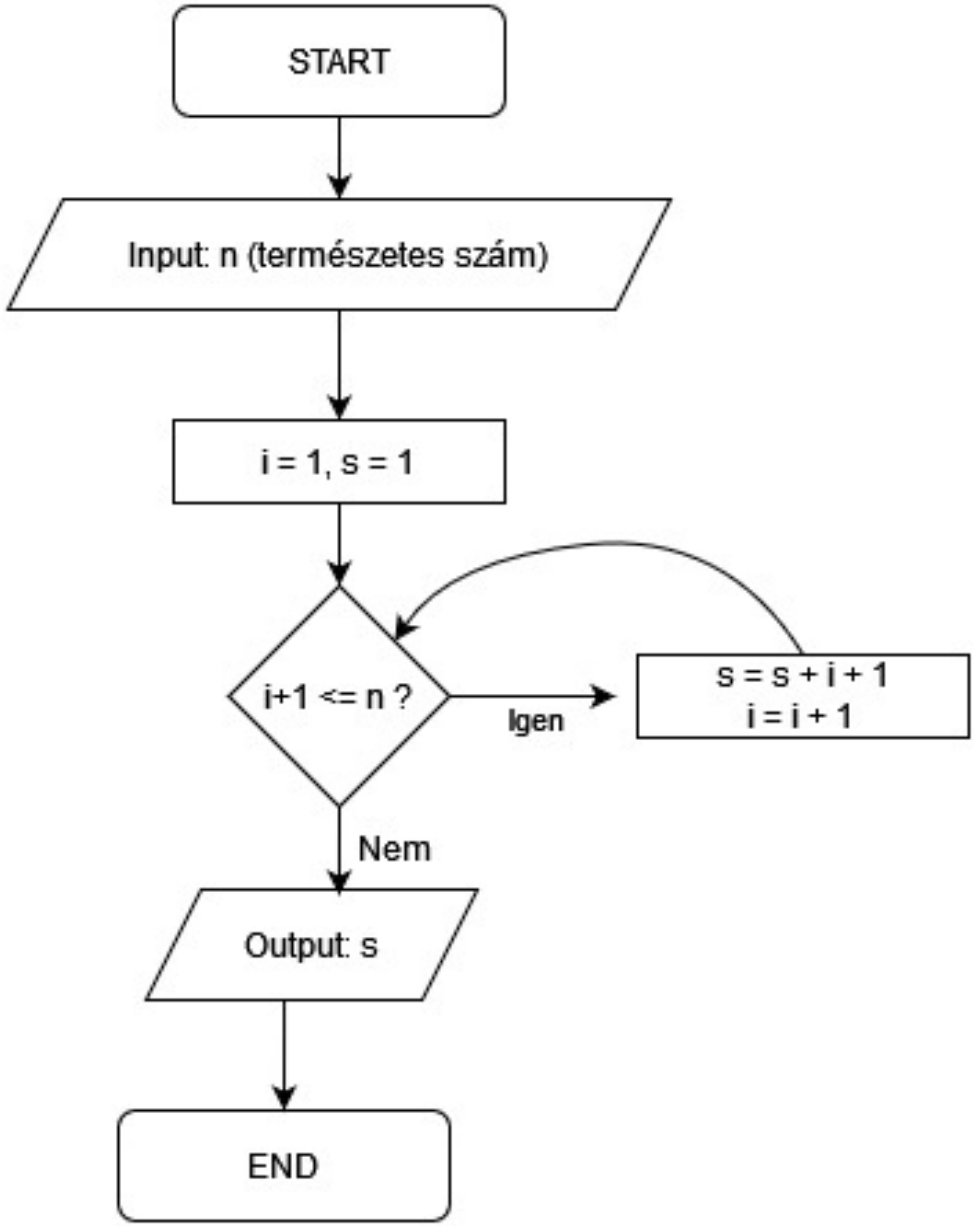
ANSI/ISO Shape	Name	Description
	Flowline (Arrowhead)[15]	Shows the process's order of operation. A line coming from one symbol and pointing at another.[14] Arrowheads are added if the flow is not t
	Terminal[14]	Indicates the beginning and ending of a program or sub-process. Represented as a stadium,[14] oval or rounded (fillet) rectangle. They usua a process, such as "submit inquiry" or "receive product".
	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 yes/no que
	Input/Output[15]	Indicates the process of inputting and outputting data,[15] as in entering data or displaying results. Represented as a parallelogram.[14]
	Annotation[14] (Comment)[15]	Indicating additional information about a step in the program. Represented as an open rectangle with a dashed or solid line connecting it to t
	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][18]
	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



...és *n* lépéssel



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

def osszeadas(n):
    sum = n*(n+1)/2
    return sum

print(linear_search([7,8,9,6,4], 9, 5))
print(osszeadas(30))
```

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 nő 1-gyel, akkor az algoritmus lépés száma nő legfeljebb 1-gyel. Az algoritmus futási ideje legfeljebb n.

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

def osszeadas(n):
    sum = n*(n+1)/2
    return sum

print(linear_search([7,8,9,6,4], 9, 5))
print(osszeadas(30))
```

```
linear_search
C:\Users\carol\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\carol\PycharmP
2
465.0
```

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 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 unsuccesful
...
>>>
>>> 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
```