2. gyakorlat

Téma:

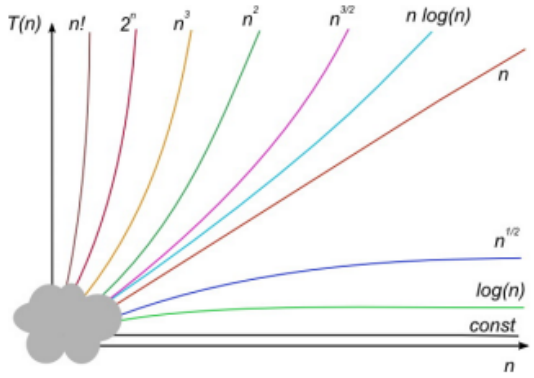
Nevezetes nagyságrendek, beszúró és összefésülő rendezések lejátszása, algoritmusok megbeszélése, verem típus, verem alkalmazása.

Nevezetes nagyságrendek:

Példák különféle műveletigényű algoritmusokra:

- $\bullet\Theta(1)$: Verem vagy sor bármely művelete
- $\bullet\Theta(\log n)$: Bináris keresés, Legendre-algoritmus
- $\bullet\Theta(\sqrt{n})$: Prímszámteszt
- $\bullet\Theta(n)$: Lineáris keresés, maximum kiválasztása
- $\bullet\Theta(n \log n)$: Gyorsrendezés, összefésülő rendezés, kupacrendezés
- $\bullet\Theta(n^2)$: Beszúró rendezés, buborékrendezés
- $\bullet\Theta(n^3)$: Mátrixszorzás
- $\bullet\Theta(2^n)$: Hanoi tornyai
- $\bullet\Theta(n!)$: Utazóügynök-probléma

Szemléltessük a nagyságrendeket az alábbi grafikon segítségével!

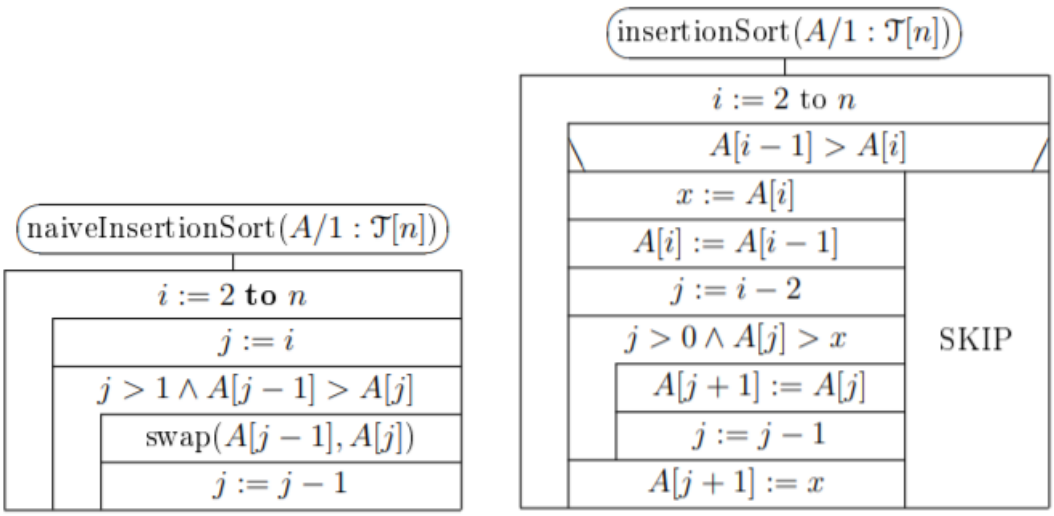


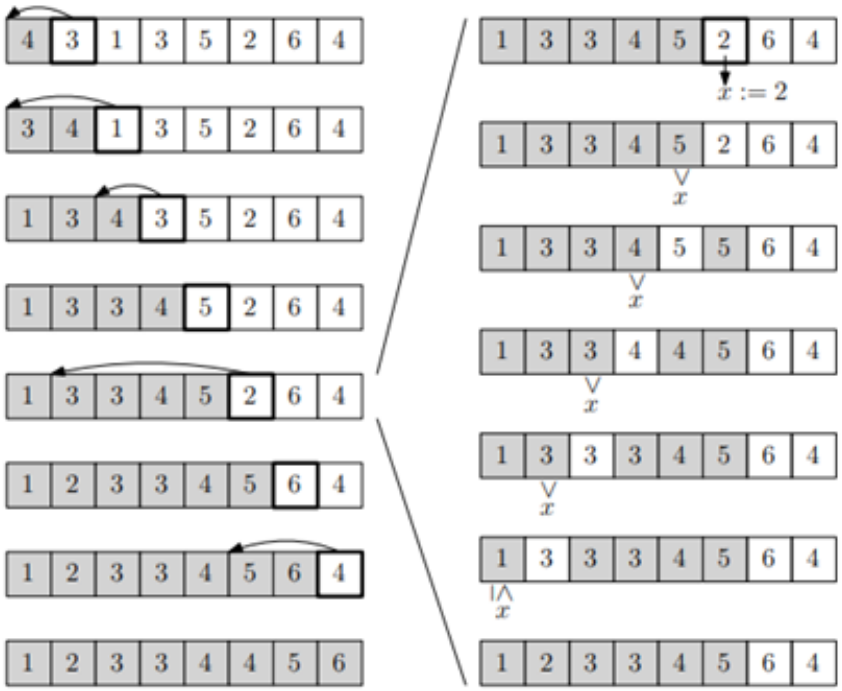
Táblázatos érzékeltetés:

n	log(n)	n^(1/2)	n	n log(n)	n^(3/2)	n^2	n^3	2^n	n!
...									
10	3,32	3,16	10	33,22	31,62	100	1000	1024	3628800
11	3,46	3,32	11	38,05	36,48	121	1331	2048	39916800
12	3,58	3,46	12	43,02	41,57	144	1728	4096	479001600
13	3,70	3,61	13	48,11	46,87	169	2197	8192	6,227E+09
...									
29	4,86	5,39	29	140,88	156,17	841	24389	536870912	8,842E+30
30	4,91	5,48	30	147,21	164,32	900	27000	1,074E+09	2,653E+32
31	4,95	5,57	31	153,58	172,60	961	29791	2,147E+09	8,223E+33
32	5	5,66	32	160	181,02	1024	32768	4,295E+09	2,631E+35
...									
64	6	8	64	384	512	4096	262144	1,845E+19	1,269E+89
...									
128	7	11,31	128	896	1448,15	16384	2097152	3,403E+38	3,86E+215
...									
256	8	16	256	2048	4096	65536	16777216	1,158E+77	****
...									
512	9	22,63	512	4608	11585,24	262144	134217728	1,34E+154	****
...									
1024	10	32	1024	10240	32768	1048576	1,074E+09	****	****

1. ábra. Forrás: Fekete István jegyzete: `https://people.inf.elte.hu/fekete/algoritmusok_jegyzet/`

Beszúró rendezés (Insertion Sort):





Stabilitás: Egy rendező eljárást stabilnak nevezünk, ha nem változtatja meg az egyenlő kulcsú elemek egymáshoz viszonyított sorrendjét. Figyeljük meg, hogy a beszúró rendezés stabil! (A rendezett szakaszba való beszúráskor a beszúrandó elem helyét jobbról balra keresi meg, és a vele egyenlőket már nem lépi át.)

Feladat: Gondoljuk át, hogy az előző gyakorlaton tanult két rendezés (buborék, maximum-kiválasztásos) stabil-e! Ha igen, miért? Ha nem, adjunk ellenpéldát, gondoljuk meg, hogyan lehetne stabillá tenni, illetve, hogy érdemes-e!

Megoldás: A buborék rendezés stabil, mert az egyenlő szomszédokat nem cseréli meg.

A maximum-kiválasztásos rendezés nem stabil, pl. (2,2',1). Ha a maximumok közül a jobb szélsőt (JMAX) választanánk, arra is adható egyszerű ellenpélda: (2,1,1'). Ha JMAX és a rendezetlen szakasz utolsó elemének cseréje helyett JMAX-ot a helyéről töröljük, a tőle jobbra levő rendezetlen elemeket balra csúsztatjuk, és JMAX-ot a rendezetlen szakasz végén keletkező üres helyre beszúrjuk, stabil lesz, de az adatmozgatások várható és maximális száma ezzel $\Theta(n^2)$-re romlik, így ez a rendezés minden előnyét elveszti.

Műveletidő: (Eljáráshívások és ciklusiterációk összege) $T_S(n)$ // S-algoritmus; *n*-input mérete

$$mT_{IS}(n) = 1 + (n-1) = n$$

(Egy eljáráshívás + a külső ciklus $(n-1)$ iterációja.)

$$MT_{IS}(n) = 1 + (n-1) + \sum_{i=2}^{n}(i-2) = n + \sum_{j=0}^{n-2} j = n + \frac{(n-1)*(n-2)}{2}$$

$$MT_{IS}(n) = \frac{1}{2}n^2 - \frac{1}{2}n + 1$$

$$AT_{IS}(n) \approx 1 + (n-1) + \sum_{i=2}^{n}\left(\frac{i-2}{2}\right) = n + \frac{1}{2} * \sum_{j=0}^{n-2} j =$$

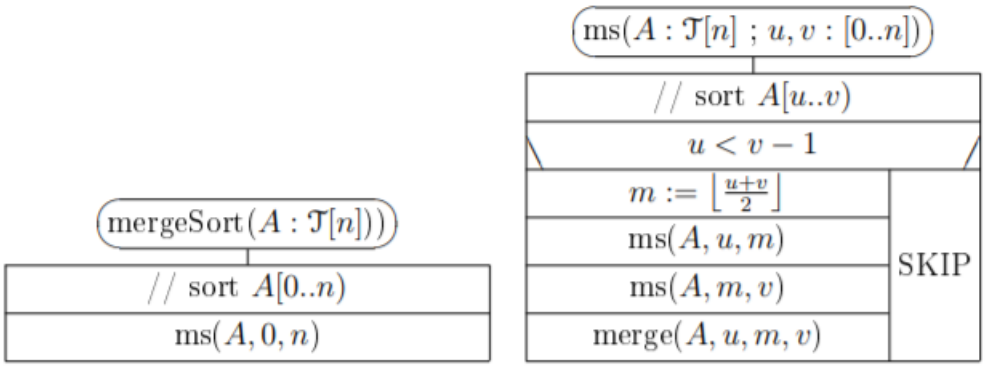
$$= n + \frac{1}{2} * \frac{(n-1)*(n-2)}{2} = \frac{1}{4}n^2 + \frac{1}{4}n + \frac{1}{2}$$

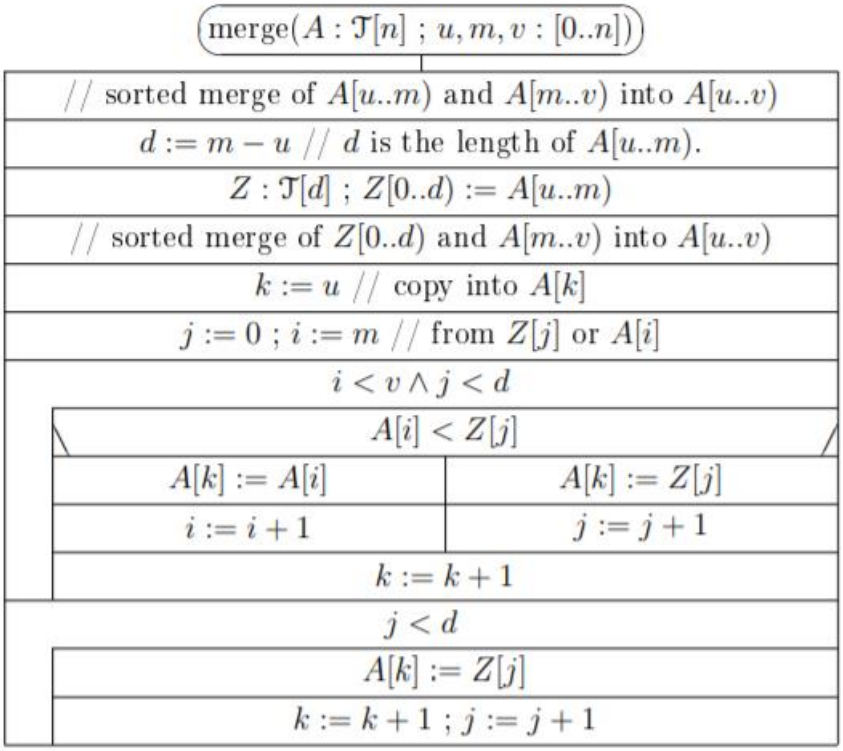
Összegezve az eredményeket:

$$mT_{IS}(n) \in \Theta(n)$$

$$AT_{IS}(n), MT_{IS}(n) \in \Theta(n^2)$$

Összefésülő rendezés (Merge Sort):





Műveletigény:

A merge(*A,u,m,v*) eljárás műveletigényének meghatározásához vezessük be az *l=v-u* jelölést.

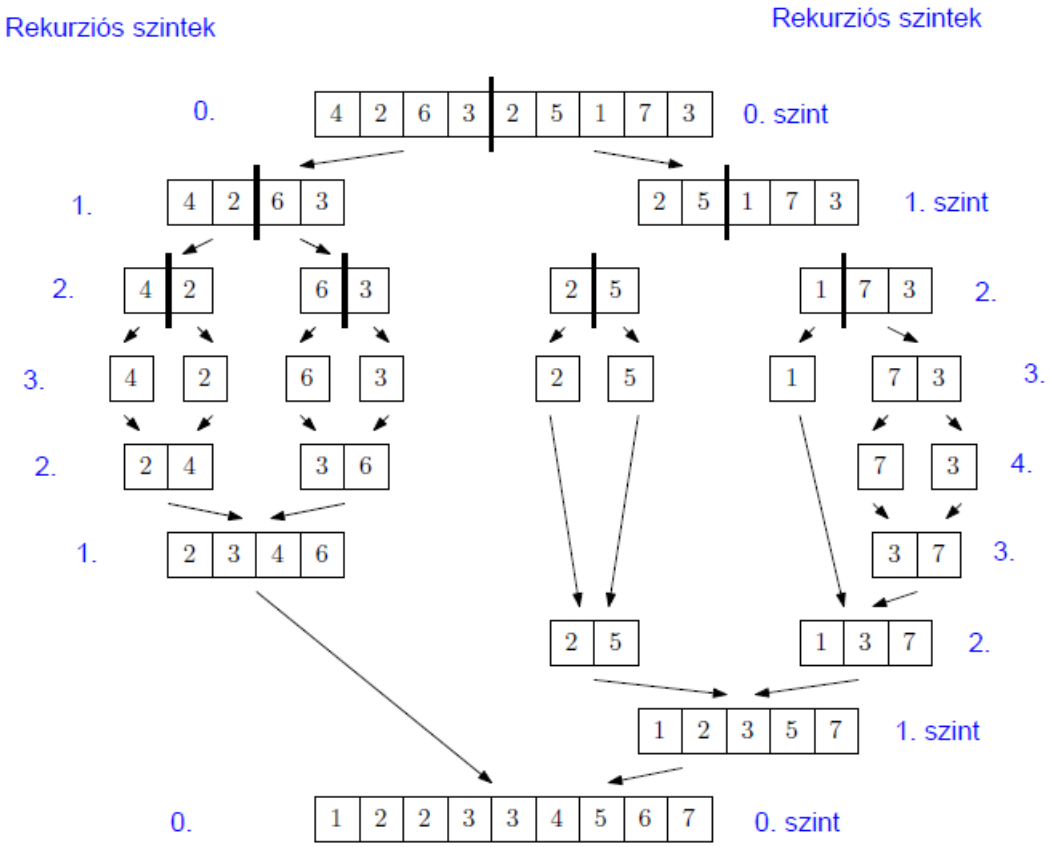
$$mT_{merge}(l) \geq 1 + \left\lfloor \frac{l}{2} \right\rfloor + \left\lfloor \frac{l}{2} \right\rfloor \geq \left\lceil \frac{l}{2} \right\rceil + \left\lfloor \frac{l}{2} \right\rfloor = l, \text{ valamint}$$
$$MT_{merge}(l) \leq 1 + \left\lceil \frac{l}{2} \right\rceil + l \leq 2l, \text{ ahonnét}$$

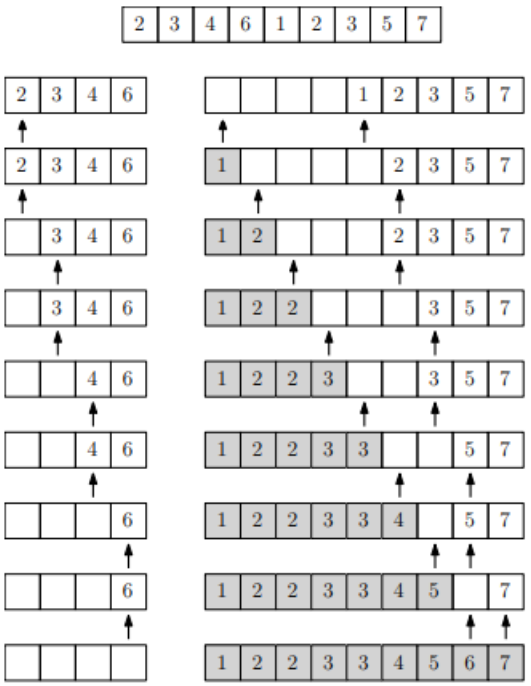
$$l \leq mT_{merge}(l) \leq MT_{merge}(l) \leq 2l$$

(Innét $mT_{merge}(l), MT_{merge}(l) \in \Theta(l)$ adódik.)

$$MT_{MS}(n), mT_{MS}(n) \in \Theta(n \log n)$$

Szemléltetés:



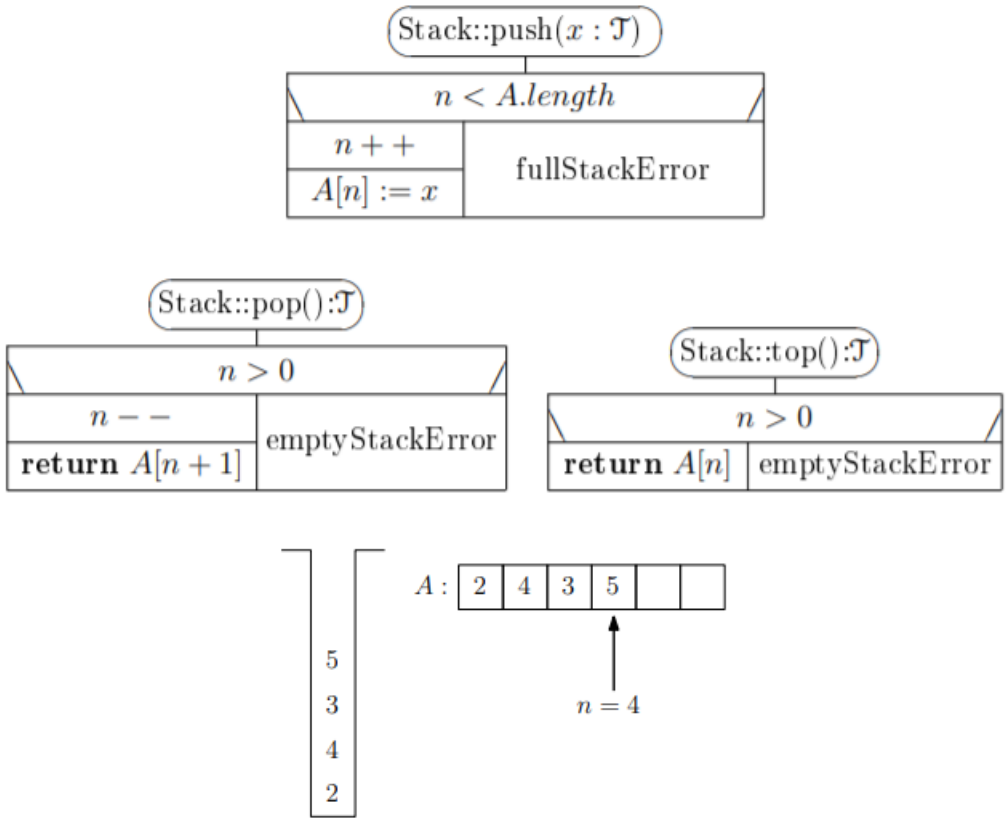


Verem (LIFO) **adattípus:**

A vermet most dinamikus tömb ($A/1 : \mathcal{T}[]$) segítségével reprezentáljuk, ahol $A.length$ a verem maximális mérete, $\mathcal{T}$ a verem elemeinek típusa.

Stack
- $A/1 : \mathcal{T}[]$ // $\mathcal{T}$ is some known type ; $A.length$ is the max. size of the stack - $n : \mathbb{N}$ // $n \in 0..A.length$ is the actual size of the stack

+ Stack($m : \mathbb{N}$) {$A :=$ **new** $\mathcal{T}[m]$; $n := 0$} // create an empty stack
+ ~ Stack() { **delete** A }
+ push($x : \mathcal{T}$) // push x onto the top of the stack
+ pop() : $\mathcal{T}$ // remove and return the top element of the stack
+ top() : $\mathcal{T}$ // return the top element of the stack
+ isFull() : $\mathbb{B}$ {**return** $n = A.length$}
+ isEmpty() : $\mathbb{B}$ {**return** $n = 0$}
+ setEmpty() {$n := 0$} // reinitialize the stack

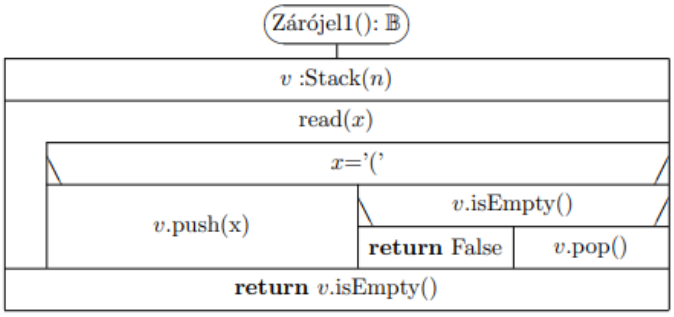


2. ábra. A verem rajzos szemléltetése

Veremmel megoldható feladatok:

1. Adott egy zárójelekből álló, legfeljebb n hosszú karaktersorozat a bemeneten. Olvassuk be, és döntsük el róla, hogy helyes zárójelezést határoz-e meg! (Vagyis, hogy párbaállíthatók-e a zárójelek úgy, hogy minden nyitó zárójelnek egy olyan csukó zárójel a párja, amely később következik a sorozatban.)

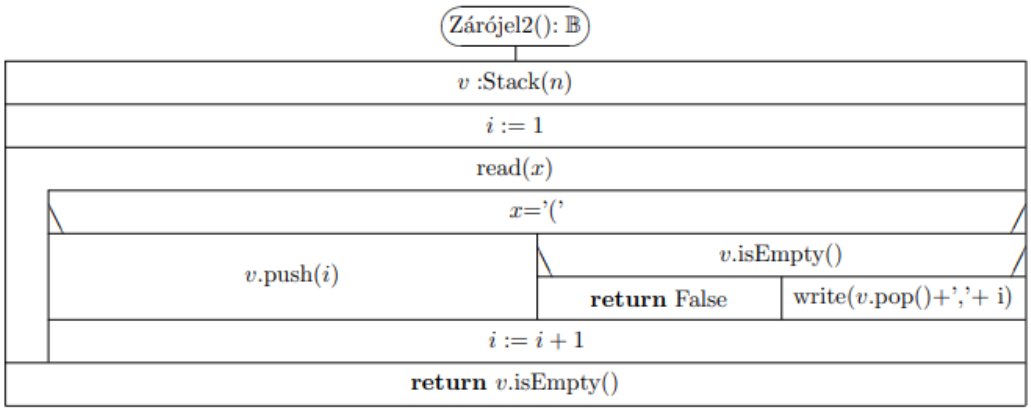
Megoldás: Verem segítségével:



A beolvasáshoz az előadásjegyzetben is használt read($\&x : \mathcal{T}$) : $\mathbb{B}$ függvényt használtuk, de ettől eltérhetünk. Ez a függvény beolvassa x-be a következő értéket a bemenetről, és ha ez sikeres, akkor igaz, ha pedig végére értünk az inputnak, akkor hamis lesz.

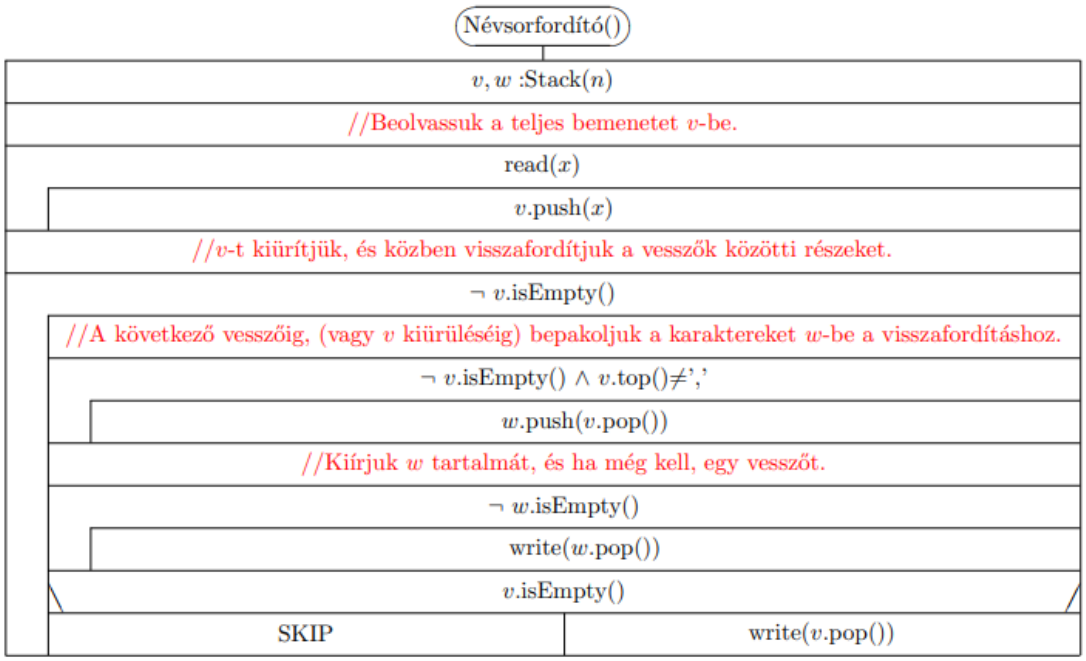
Vegyük észre, hogy a fenti módszerhez valójában nincs is szükség verem használatára. Elég lenne egyetlen számláló változóval nyilvántartani azt, hogy eddig mennyivel több nyitó zárójellel találkoztunk, mint csukó zárójellel. Módosítsuk úgy a feladatot úgy, hogy tényleg szükség legyen a veremre: Ezúttal írassuk ki az összetartozó zárójelpárok indexeit!

2. Adott egy zárójelekből álló, legfeljebb n hosszú karaktersorozat a bemeneten. Olvassuk be, és döntsük el róla, hogy helyes zárójelezést határoz-e meg! Írassuk ki az összetartozó zárójelpárok indexeit!



3. A bemenetről karakterenként beolvasunk egy (legfeljebb n karakterből álló) névsort, ahol a nevek egymástól vesszővel vannak elválasztva. Írjuk ki a neveket fordított sorrendben!
Például: András,Béla,Csaba $\mapsto$ Csaba,Béla,András

Megoldás:



Javasolt házi feladat

Tükrözött-e a szöveg? Adott egy legfeljebb n hosszú, betűkből és '#' szimbólumokból álló sorozat. A sorozatot tükrözöttnek nevezzük, ha felbontható olyan páratlan hosszú, palindrom karaktersorozatokból álló részekre, amelyeknek középső karaktere az egyetlen bennük szereplő '#'. Döntsük el a bemenetről olvasott szövegről, hogy tükrözött-e!

Példa:
Tükrözött: #, ####, abc#cba, ##a#aabc#cba
Nem tükrözött: abc, abc#cb, abc#cbaa#aa, ab#bac##c

Egy lehetséges megoldás: