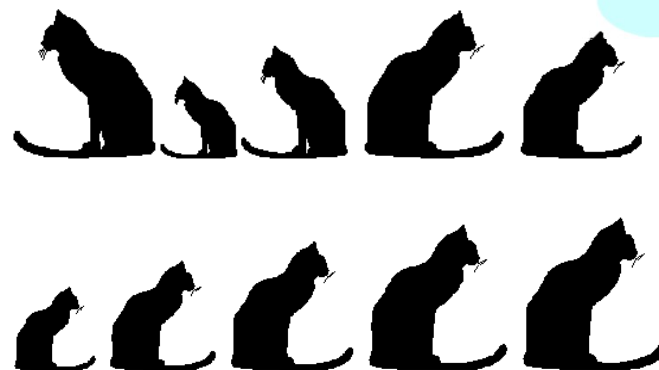


□ Rendezés minimális elem kiválasztással

- *Az algoritmus működése*
- *Rávezető feladat*
- *A rávezető feladat programja*
- *A rendezés Chapin diagramja*
- *A rendezés programja*



□ Az algoritmus működése



Rendezés: célunk egy adott számsorozat növekvő sorrendbe rendezése.

Példa:

Adott a rendezendő sorozat:

[7,8,4,9,1]

Rendezés után ezt várjuk:


[1,4,7,8,9].

A minimális elem kiválasztásával működő algoritmus

jobbra követhető:

	$i =$	0	1	2	3	4
$j = 0$		7	8	4	9	1
		4	8	7	9	1
		1	8	7	9	4
$j = 1$		1	8	7	9	4
		1	7	8	9	4
		1	4	8	9	7
$j = 2$		1	4	8	9	7
		1	4	7	9	8
		1	4	7	9	8
$j = 3$		1	4	7	8	9
		1	4	7	8	9
	Eredmény:	1	4	7	8	9

?
 $V[j] > V[i]$
 j. és i. elemek összehasonlítása

 : már rendezett elemek



□ **Az algoritmus működése ..**

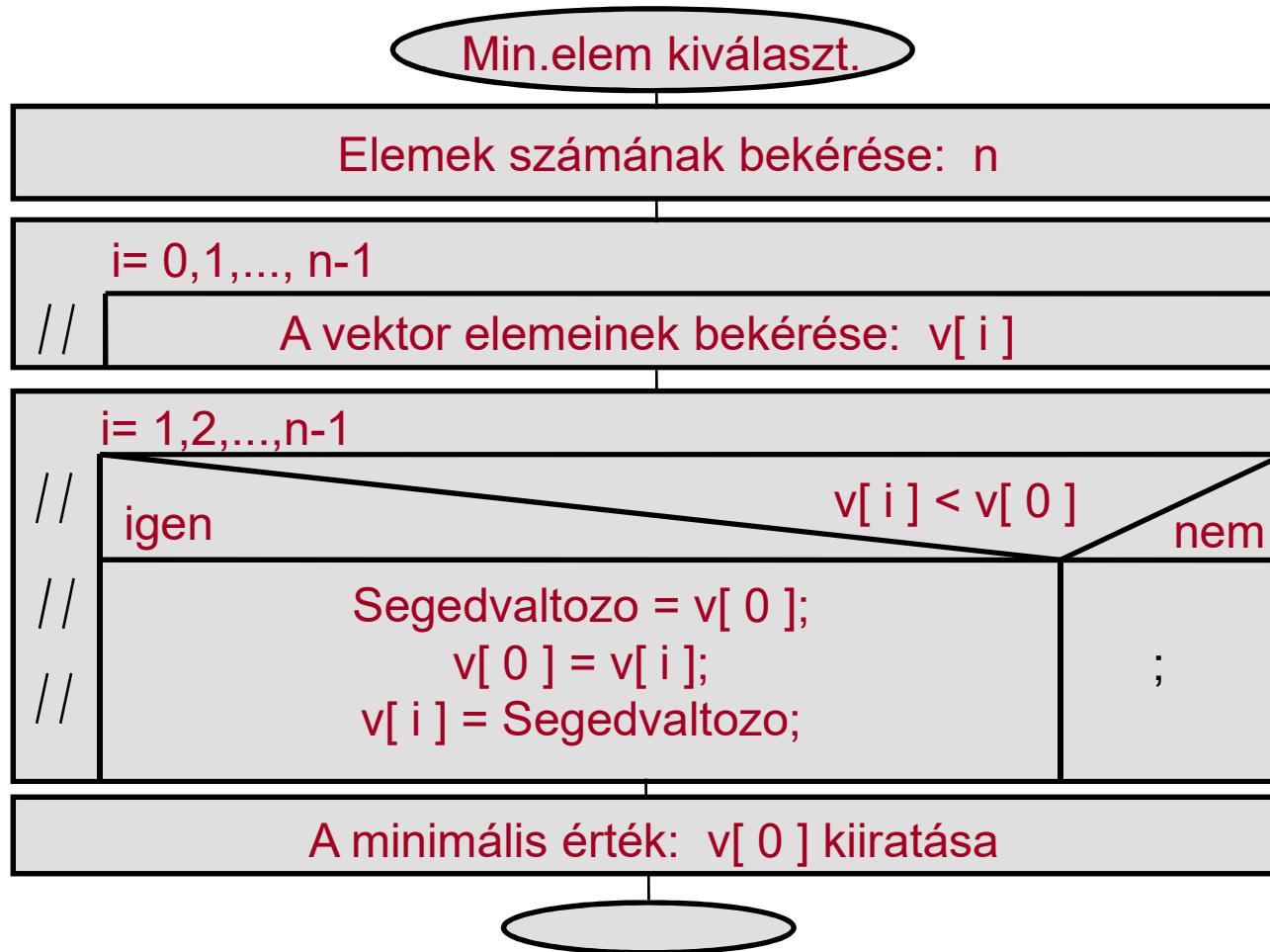
a még rendezetlen részsorozat minimális elemét a részsorozat első pozíciójába visszük elemcserével. Két elem cseréjéhez egy segédváltozóra is szükség van, ahová kimenekíthetjük a $v[j]$ elemet, amelynek a helyére a $v[i]$ elemet akarjuk írni. Miközben i értéke végigfut $j+1$ -től 4-ig, minden $v[i] < v[j]$ párnál cserét hajtunk végre.

□ **Rávezető feladat:**

Alakítsuk át a korábban megismert minimális elem kiválasztó algoritmusunkat úgy, hogy a minimális érték a sorozat elemeit tartalmazó vektor 0. indexű helyén adódjon !

A Chapin diagram 

□ **A rávezető feladat struktogramja**



□ **A rávezető
feladat
programja:**

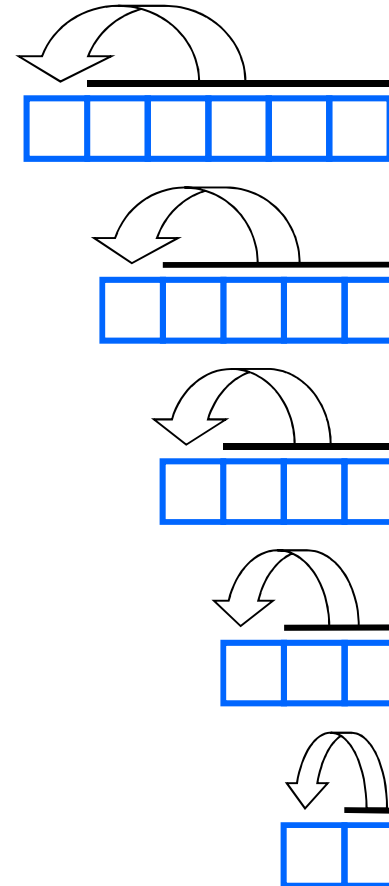
```
#include <stdio.h>
int v[100];
void main()
{ int i,n,Segedvaltozo;
  printf("Minimális elem kiválasztás\n");
  printf("Elemek száma="); scanf("%d",&n);
  for (i=0; i <n; i++)
  {
    printf("v[ %d ]=",i ); scanf("%d",&v[ i ]);
  }
  for (i=1; i < n; i++)
  {
    if ( v[ i ] < v[0])
    {
      Segedvaltozo = v[0];    /* csere */
      v[0] = v[ i ];
      v[ i ] = Segedvaltozo;
    }
  }
  printf("\nA minimális elem: v[0]= %d", v[0]);
}
```



□ **A rendezés Chapin diagramja:**

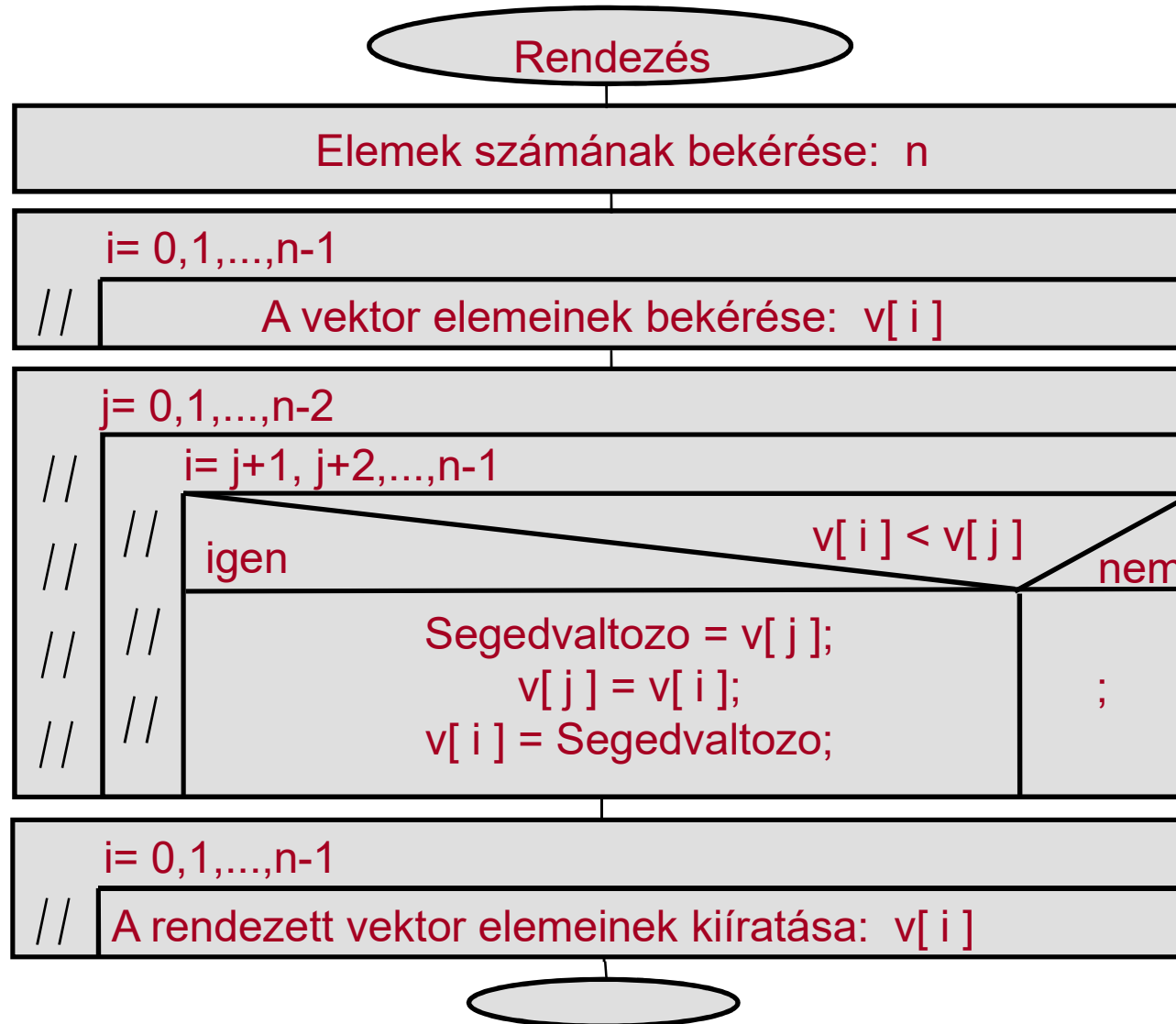
Mivel egyre rövidülő részsorozatok minimális elemének az aktuális részsorozat első pozíciójába vitele a célunk, az egyetlen sorozaton működő előző algoritmust ciklusba kell ágyazni.

*Figyeljük majd meg, hogy a belső **for** ciklus kezdőértékét a külső ciklusváltozó aktuális értéke határozza meg.*



A megoldás a következő diagramon látható: 

□ **A rendezés Chapin diagramja ..**



□ **A rendezés programja:**



```
#include <stdio.h>
int v [100];
void main()
{
    int i, j, n, Segedvaltozo;
    printf("Rendezés minimális elem kiválasztással\n");
    printf("Elemek száma="); scanf("%d",&n);
    for (i =0; i < n; i++)
    {
        printf("v [ %d ]=", i+1); scanf("%d",&v[ i ]);
    }
}
```



□ **A rendezés programja ..**

```
for (j = 0; j < n - 1; j++)  
{  
    for (i = j + 1; i < n; i++)  
    {  
        if (v[i] < v[j])          /* csere */  
        {  
            Segedvaltozo = v[j];  
            v[j] = v[i];  
            v[i] = Segedvaltozo;  
        }  
    }  
}  
printf("\n A rendezett vektor elemei: ");  
for (i = 0; i < n; i++)  
{  
    printf(" v [ %d ] = %d \n", i+1, v[i] );  
}  
}
```

