

findLongestSeq (v[], n, *nr)	Cost	Nr repetitii
int best[], vrez[], i, j, max;		
max = 0;	C1	1
for i = 0 : n-1	C2	n+1
best[i] = 1;	C3	n
for i = 1 : n - 1	C4	n
for j = 0 : i - 1	C5	sum(i)+1, i= 1:n-2
if (v[i] > v[j] && best[i] < best[j] + 1)	C6	sum(i)+1
best[i] = best[j] + 1;	C7	x
if (max < best[i])	C8	x
max = best[i];	C9	y
*nr = max;	C10	1
for i = n -1 : 0	C11	n+1
if (best[i] == max)	C12	n
vrez[max-1] = v[i];	C13	z
max--;	C14	z
return vrez;	C15	1

În cazul cel mai defavorabil, operația critică este operația C6, care se execută de $(n-1) \cdot \text{sum}(i)+1$, cu $i = 1 : n-2$.

$$T(n) = \sum_{i=1}^{n-2} (i) + 1 = \frac{(n-1)(n-2)}{2} + 1 = O(n^2)$$

Descrierea algoritmului:

Se folosește un vector adițional best în care sunt păstrate lungimile celei mai lungi secvențe de numere crescătoare de pe poziția curentă. Se păstrează lungimea celei mai lungi secvențe și se parcurge vectorul de la sfârșit. Se adaugă într-un vector rezultat cea mai lungă secvență.