1. Procedure SortingMax(In / Out L:List)

Kamus

Max1, Max2 : adr

Count : Integer

Procedure deleteFirst(List, Adr)

Procedure insertFirst(List, Adr)

Procedure insertLast(List, Adr)

Procedure insertAfter(List, Prec, Adr)

Function findMax(List) → Adr

Algoritma

Max1 = findMax(L1)

If L.first == Max1 then

deleteFirst(L, P)

else

Q = L.first

While Q ->next != Max1 do

Q = Q->next

Endwhile

If Max->next != NIL then

Q->next = Max1->next

Endif

Max1->next = NIL

Endif

Max2 = findMax(L1)

If L.first == Max2 then

deleteFirst(L, P)

else

Q = L.first

While Q ->next != Max2 do

Q = Q->next

Endwhile

If Max->next != NIL then

Q->next = Max2->next

Endif

Max2->next = NIL

Endif

If Max1->info == Max2->info then

Count = 0

Q = L.first

While Q != NIL do

Count = Count + 1

Q = Q->next

Endwhile

Count = Count div 2

Q = L.first

For i = 1 to Count do

Q = Q->next

Endfor

insertAfter(L, Q, Max 1)

insertAfter(L, Max1, Max2)

Else

inserFirst(L, Max1)

inserLast(L, Max2)

Endif

Endprocedure

2. Procedure SwapMinMax(In / Out L:List)

Kamus

Min, Max : Adr

Temp : Integer

Function findMin(List) → Adr

Function findMax(List) → Adr

Algoritma

Min = findMin(L)

Max = findMax(L)

Temp = Min->info

Min->info = Max->info

Max->info = Temp

endprocedure