Algoritmusok

Keresés algoritmusa

Be: t[] lista, egész i(lista elemszáma), Ttul()

Ki: i

Ciklus amíg (i<t hossz és t[i] nem egyenlő Ttul())

i++;

ciklus vége

Ha(i<t hossz)

Ki: i

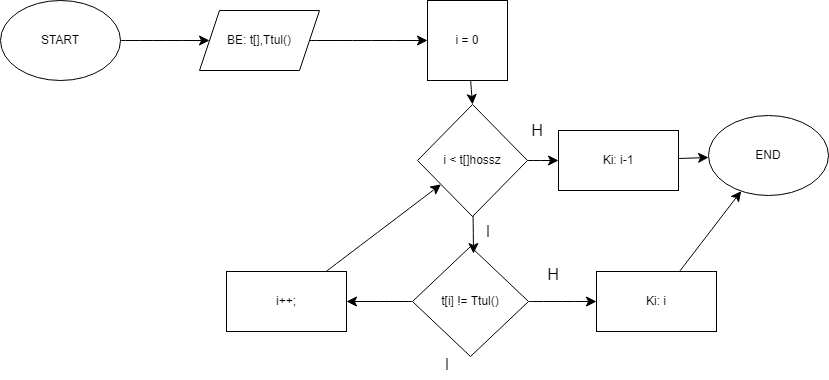
elágazás vége

Ellenben

Ki: i - 1

elágazás vége

progam vége;



Maximum kiválasztás index

Be: t[] lista, egész i, maxi

Ki: maxi

maxi = 0

ciklus i = 1-től t hosszig egyesével

ha(t[maxi]<t[i])

maxi = i

elágazás vége

ciklus vége

Ki: maxi

program vége

A képen diagram, Műszaki rajz, Tervrajz, vázlat látható

Automatikusan generált leírás

Egyszerű cserés rendezés

Be: t[]lista, egész i, egész j, egész csere

Ki:t[]lista

ciklus i = 0-től t[] hosszig -2

ciklus j = i+1 től t[] hosszig -1

ha t[i]<t[j]

csere = t[i]

t[i] = t[j]

tfj] = csere

e.v.

c.v.

c.v.

Ki: t[]

A képen diagram, Tervrajz, sor, Műszaki rajz látható

Automatikusan generált leírásp.v.

Be: A lista, B lista, n: (A elemszáma), m: (B elemszáma)

Előfeltétel: A és B lista nem tartalmaz ismétlődő elemet

Ki: metszet lista

Utófeltétel: Az A és B listák közös eleme (Metszet)

ciklus n=0-tól a[]hosszig

ciklus m=0-tól b[]hosszig

ha(a[n] = b[m])

metszet = metszet.add( a[n])

e.v

c.v

c.v

Ki: metszet

p.v

A képen diagram, Tervrajz, Műszaki rajz, sor látható

Automatikusan generált leírás

Be: A lista, B lista, n: (A elemszáma), m: (B elemszáma)

Előfeltétel: A és B lista nem tartalmaz ismétlődő elemet

Ki: C lista

ciklus i = 1 .. n

C[i] = A[i]

ciklus vége

k = n

ciklus j = 1 .. m

i = 1

ciklus amíg i<=n és b[j]<>a[i]

i = i + 1

ciklus vége

ha i>n akkor

C[k] = b[j]

k = k + 1

ciklus vége