Seminar 1 – Recursivitate

1. Algorimti recursivi cu numere

1.1. Se da un numar natural nenul.

Se cere sa se verifice daca numarul dat este “norocos”. Un numar se considera norocos daca este format exclusiv din cifrele 2 si 4.

Ex: 224 – numar norocos

225 – nu este norocos

Model matematic recursiv:

Norocos(n) = { adevarat , n = 0

{ fals , n > 0 si n%10 != 2 si n%10 !=4

{ norocos([n/10]) , altfel

Norocos(n) = { adevarat , n = 0

{ (n%10 = 2 sau n%2 =4) si norocos([n/10]) , n > 0

Implementare Python:

Def norocos(n):

If n == 0:

return True

If n%10 != 2 and n%10 != 4:

return False

Return norocos(n//10)

1.2. Se da un numar natural n > 1.

Se cere sa se calculeze suma divizorilor proprii si improprii ai lui n.

Ex: n = 6 => 1 + 2 + 3+ 6 = 12

Sumadiv(n,s,d) = { s + d, d = n

{ sumadiv(n,s+d,d+1), d<n, n%d=0

{ sumadiv(n,s,d+1), altfel

SumadivPrincipal(n) = sumadiv(n,0,1)

Suma\_div(n, d) = { d, d = n

{d + suma\_div(n, d + 1), d < n, n % d = 0

{suma\_div(n, d + 1), altfel

Suma\_principala(n) = suma\_div(n, 1)

Def suma\_div(n,d)

If n==d:

return d

If n%d==0:

return d + suma\_div(n, d+1)

Else

return suma\_div(n, d+1)

1. Algoritmi recursivi cu liste

L1l2...ln

|  |  |
| --- | --- |
| DA | NU |
| N=0? N = 1, n > 1, n =2 ? | N, n = k (k var.), n par?, n impar? |
| L1 , l2, l3 | Lk (k var.), ln |
| L2...ln, l3...ln | Lk...ln (k var) |
| Inserare la inceput e (+) l1l2...ln / el1l2...ln | Inserare pe o pozitie oarecare |

2.1. Se da o lista numerica.

Se cere sa se calculeze **produsul elementelor pare**.

Produs(l1l2...ln) = { 1 , n=0

{ l1 \* produs(l2l3...ln), n > 0 si l1%2 = 0

{ produs(l2l3...ln), altfel

Vida(Lista): True/False

Sublista(Lista): Lista (l2...ln)

PrimElement(Lista): Int

Def produs(l):

If vida(l) == True:

return 1

If PrimElement(l) % 2 == 0:

return PrimElement(l) \* produs(Sublista(l))

Return Produs(sublista(l))

2.2. Se da o lista l, un element e si o pozitie m > 0.

Se cere sa se insereze in lista l elementul e pe pozitia m.

Ex: l = [1,2,3,4,5,6]

E = 4

M = 2

=> [1,4,2,3,4,5,6]

Insereaza(l1l2l3...ln, m, e) = { (e), n=0 si m = 1

{ (), m > 1 si n= 0

{ e + l1l2...ln , m=1 si n >= 0

{ l1 + insereaza(l2l3...ln, m-1, e), n > 0 si m>1

2.3. Se da o lista l, un element e si o valoare m > 1.

Se cere sa se insereze in lista l elementul e din m in m.

Ex: l = [1,2,3,4,5,6,7,8,9,10 ,11, 12]

E = 0

M = 4

=> [1, 2, 3, 0, 4, 5, 6, 0, 7, 8, 9, 0, 10, 11, 12, 0]

Insereaza(l1l2l3...ln, m,mi, e) = { (e), n=0 si m = 1

{ (), m > 1 si n= 0

{ e + insereaza(l1l2...ln, mi, mi, e) , m=1 si n >= 0

{ l1 + insereaza(l2l3...ln, m-1, e), n > 0 si m>1

Insereaza(l1l2l3...ln, m,c, e) = { (e), n=0 si c%m=0

{ (), c%m != 0 si n= 0

{ e + insereaza(l1l2...ln, m, c+1 e) , c%m=0 si n >= 0

{ l1 + insereaza(l2l3...ln, m,c+1, e), n > 0 si c%m != 0