

Redactare și comunicare științifică și profesională

Laborator 5

Lect. dr. Adela Sasu

November 11, 2020

Exercițiu 1:

Inserați trei imagini apoi decupați din fiecare o parte care vi se pare interesantă. De exemplu, în figura de mai jos decupăm veverița:



Figura 1:

Realizați o figură în care să includeți trei poze după modelul:



(a) Toamna 1



(b) Toamna 2



(c) Toamna 3

Figura 2: Toamna

Algoritmul lui Euclid se redactează în L^AT_EX

```
\begin{algorithm}
\caption{Algoritmul lui Euclid}\label{euclid}
\begin{algorithmic}[1]
\Procedure{Euclid}{$a,b$}\Comment{C.m.m.d.c al lui a si b}
\State{$r\gets a\bmod b$}
\While{$r\neq 0$}\Comment{Avem raspunsul daca $r=0$}
\State{$a\gets b$}
\State{$b\gets r$}
\State{$r\gets a\bmod b$}
\EndWhile\label{euclidendwhile}
\State{\textbf{return} $b$}\Comment{C.m.m.d.c este b}
\EndProcedure
\end{algorithmic}
\end{algorithm}
```

iar prin compilare se afișează

Algorithm 1 Algoritmul lui Euclid

```
1: procedure EUCLID( $a, b$ )                                ▷ C.m.m.d.c al lui a si b
2:    $r \leftarrow a \bmod b$ 
3:   while  $r \neq 0$  do                                       ▷ Avem raspunsul daca  $r = 0$ 
4:      $a \leftarrow b$ 
5:      $b \leftarrow r$ 
6:      $r \leftarrow a \bmod b$ 
7:   end while
8:   return  $b$                                               ▷ C.m.m.d.c este b
9: end procedure
```

Exercițiu 2:

Redactați următorii algoritmi:

```

procedure insert( $T[1 \dots n]$ )
  for  $i \leftarrow 2$  to  $n$  do
     $x \leftarrow T[i]; j \leftarrow i-1$ 
    while  $j > 0$  and  $x < T[j]$  do
       $T[j+1] \leftarrow T[j]$ 
       $j \leftarrow j-1$ 
     $T[j+1] \leftarrow x$ 

procedure select ( $T[1 \dots n]$ )
  for  $i \leftarrow 1$  to  $n-1$  do
     $minj \leftarrow i; minx \leftarrow T[i]$ 
    for  $j \leftarrow i+1$  to  $n$  do
      if  $T[j] < minx$  then
         $minj \leftarrow j$ 
         $minx \leftarrow T[j]$ 

     $T[minj] \leftarrow T[i]$ 
     $T[i] \leftarrow minx$ 

```

Figura 3: Algoritmi elementari de sortare

```

function greedy( $C$ )
  { $C$  este mulțimea candidaților}
   $S \leftarrow \emptyset$  { $S$  este mulțimea în care construim soluția}
  while not soluție( $S$ ) and  $C \neq \emptyset$  do
     $x \leftarrow$  un element din  $C$  care maximizează/minimizează select( $x$ )
     $C \leftarrow C \setminus \{x\}$ 
    if fezabil( $S \cup \{x\}$ ) then  $S \leftarrow S \cup \{x\}$ 
  if soluție( $S$ ) then return  $S$ 
  else return “nu există soluție”

```

Figura 4: Algoritm Greedy

```

function russe( $A, B$ )
  arrays  $X, Y$ 
  {inițializare}
   $X[1] \leftarrow A; Y[1] \leftarrow B$ 
   $i \leftarrow 1$  {se construiesc cele două coloane}
  while  $X[i] > 1$  do
     $X[i+1] \leftarrow X[i] \text{ div } 2$  {div reprezintă împărțirea întreagă}
     $Y[i+1] \leftarrow Y[i] + Y[i]$ 
     $i \leftarrow i+1$ 
  {adună numerele  $Y[i]$  corespunzătoare numerelor  $X[i]$  impare}
   $prod \leftarrow 0$ 
  while  $i > 0$  do
    if  $X[i]$  este impar then  $prod \leftarrow prod + Y[i]$ 
     $i \leftarrow i-1$ 
  return  $prod$ 

```

Figura 5: Algoritmul înmulțirii "a la russe"

Un algoritm pentru șirul Fibonacci:

```

function fib3( $n$ )
   $i \leftarrow 1; j \leftarrow 0; k \leftarrow 0; h \leftarrow 1$ 
  while  $n > 0$  do
    if  $n$  este impar then
       $t \leftarrow jh$ 
       $j \leftarrow ih + jk + t$ 
       $i \leftarrow ik + t$ 

     $t \leftarrow h^2$ 
     $h \leftarrow 2kh + t$ 
     $k \leftarrow k^2 + t$ 
     $n \leftarrow n \text{ div } 2$ 
  return  $j$ 

```

Figura 6: Șirul lui Fibonacci