8.12.2010

Zadania przygotowawcze do drugiego kolokwium ze Wstępu do programowania imperatywnego.

Zad 1. Napisz funkcję

function Iloczyny6(var T:array[1..n]of integer; k:integer):Boolean;

która przyjmie wartość "true", wtedy i tylko wtedy gdy iloczyn liczb zapisanych w każdym segmencie tablicy T długości k jest podzielny przez 6 dla k>0.

Zad 2. Tablica A:tab zawiera albo liczby dodatnie, albo wartości -1, przy czym wszystkie wartości -1 występują w jednym spójnym segmencie. Ponadto dla dowolnych i,j takich że 1<=i<=j<=n zachodzi

$$(A[i]<>-1 \text{ oraz } A[j]<>-1) \Rightarrow (A[i]<=A[j]).$$

Napisz funkcję znajdz(const A:tab; x:Integer):1..n, , która znajdzie w tablicy A indeks zadanej liczby dodatniej x, o ile ona się tam znajduje. Jeśli liczby x nie ma w A, to wartością funkcji powinno być -1.

Zad 3. Napisz funkcję

takiesame(var A:array[1..2*N]of integer):Boolean;

która sprawdzi, czy z zapisanego w tablicy ciągu 2N liczb całkowitych można wybrać N-elementowy podciąg tak, by ciągi elementów wybranych i niewybranych były takie same.

Zad 4. Dany jest typ tab=array[1..n]of Integer, przy czym n>0. Dla tablicy A:tab oraz liczby całkowitej k powiemy, że segment A[d..g] jest k-płaski, jeśli dla każdych i,j takich, że $d \leq i,j \leq g$ zachodzi nierówność A[i]-A[j]<=k. Napisz funkcję

kaplaski(const A:tab, k:Integer):Integer,

która wyznaczy długość najdłuższego k-płaskiego segmentu w tablicy A.

Zad 5.Niech T={1,...,n}. Funkcja f: $T \rightarrow T$ jest niemalejąca, a jej wartości są zapisane w tablicy F: array[1..n] of Integer tak, że f(k)=F[k] dla k=1,...,n. Definiujemy funkcję $g_f : T \rightarrow T$ w następujący sposób:

$$g_f(k)=|\{a \in T : f(a)=k\}|,$$

gdzie przez |X| oznaczamy liczność (moc) zbioru X. Napisz fragment programu, który zmiennej rekord nada wartość równą maksimum spośród wartości przeciwobrazów funkcji g_f z jednoelementowych podzbiorów zbioru T, czyli

$$\max(|\{g_f^{-1}(\{k\})|: k=1,...,n\}).$$

Zad 6 Przez segment słowa $v \in A^*$ składający się z tych samych liter x nazwiemy takie słowo $w = x^k$ dla pewnego $x \in A$ oraz $k \in N^+$, że v=w'ww'' dla pewnych $w',w'' \in A^*$. Segment taki jest maksymalny, jeśli ani w' nie kończy się na x, ani w'' nie zaczyna się od x. Napisz gramatykę generującą wszystkie słowa v nad alfabetem {a,b,c} o tej właściwości, że każdy

maksymalny segment złożony z samych liter b w słowie v musi być bezpośrednio poprzedzony przez dłuższy segment złożony z samych liter a.

Zad 7. Dane są dwie funkcje f,g:{0,1}* $\rightarrow$ Z określone w następujący sposób:

f(ε)=0, f(0w)=g(w)+1, f(1w)=g(w)-1,

g(ε)=0, g(0w)=f(w)-1, g(1w)=f(w)+1. dla w∈{0,1}*.

Zdefiniujmy język L = {w∈{0,1}*: f(w)=0}.

Napisz funkcję

JestL(const T:array[1..n]of Integer):Boolean,

która sprawdzi, czy słowo zapisane w tablicy T należy do języka L.

Zad 8. Dana jest następująca funkcja:

```
function coto(n:Integer):Integer;
var i,x,y:Integer;
begin
  x:=0; y:=1;i:=2;
  while i<=n do begin
    x := x+y;
    y := x-y;
    i:=i+2
  end;
if odd(n) then coto := y
        else coto := x
end;
```

Określ dla każdego n, co jest wartością funkcji coto(n) i udowodnij to.

Zad 9. Dana jest następująca funkcja:

```
function coto(n:Integer):Integer;
var x,y:Integer;
begin
  x:=0; y:=1;
  while y<=n do begin
    x := x-y;
    y := y-x
  end;
if odd(n) then coto := y
```

```
        else coto := -x
end;
```

Określ dla każdego n, co jest wartością funkcji coto(n) i udowodnij to.

Zad 10. Dany jest typ tab=array[1..n] of Real, {n>0} oraz ciąg prostokątów $P_1,...,P_n$. Tablice A,B:tab reprezentują długości boków tych prostokątów tak, że dla każdego i=1,...,n para (A[i],B[i]) określa długości boków prostokąta P_i. Ciąg tych prostokątów ma tę szczególną własność, że tablica A jest posortowana niemalejąco, a tablica B nierosnąco. Napisz funkcję minprzek(const A,B:tab):Real, która obliczy długość najkrótszej przekątnej prostokątów $P_1,...,P_n$. Uwaga: funkcja pierwiastka kwadratowego ma w Pascalu nazwę sqrt.

Zad 11. Dany jest typ tab=array[1..n] of Real, {n>0}. Napisz funkcję mindist(const A,B:tab):Real, która obliczy wartość min(|A[k]-B[k]| : 1<=k<=n), przy założeniu, że tablica A jest posortowana rosnąco, a B malejąco.

Zad 12 .Podaj gramatykę języka $L=\{w \in \{a,b\}^* : \#(a,w)=2\#(b,w)\}$.

Zad 13. W tablicy A typu tab2=array[1..N] of Real jest zapisana wysokość pokrywy śnieżnej na dachu pewnego budynku w dniach 1..N i wiemy, że jest to ciąg słabo bitoniczny (najpierw niemalejący, potem nierosnący). Każdy centymetr świeżego śniegu (z danego dnia) waży nw, zaś centymetr śniegu z dnia poprzedniego jest c razy cięższy niż w dniu poprzednim. Kolejne warstwy śniegu układają się na poprzednich; jeśli śnieg topnieje to dzieje sie to w kolejności odwrotnej, czyli najpierw topnieje śnieg z wczoraj, potem z przedwczoraj itd. Napisz funkcję

zaciezko(const A:tab2; c,r,nw:Real):Boolean,

która sprawdzi, czy był taki dzień, w którym ciężar śniegu na dachu przekroczył wartość krytyczną r.

Zad 14. Niech A będzie tablica typu tab3=array[0..n-1] of Integer, gdzie n jest parzyste i większe od zera. Napisz funkcję

k_antybiton(var A:tab3):Boolean,

sprawdzającą, czy istnieje takie k z przedziału 0..n-2, że tablica B, równa tablicy A przesuniętej cyklicznie o k elementów w prawo, jest ściśle antybitoniczna, czyli od początku do pewnego miejsca malejąca i od tego samego miejsca do końca rosnąca.

Zad15. Rozważamy funkcję $f:\{1,..,n\} \to \{1,..,n\}$. Dla każdego naturalnego k określamy $f^k(x) = f(x)$ dla k=0 oraz $f^k(x)=f^{k-1}(f(x))$ dla k>0. Powiemy, że y jest zależne od x jeśli istnieje k > 0 takie, że $y=f^k(x)$. O funkcji f wiemy tyle, że wyznaczona przez nią relacja zależności jest antysymetryczna. Dwa elementy x i y są niezależne, jeśli ani x nie jest zależne od y ani y od x. Zbiór X nazwiemy niezależnym, jeśli każde dwa jego elementy są niezależne. Funkcję f definiujemy za pomocą tablicy typu tab4=array[1..n] of Integer. Napisz funkcję

maxniezalezne(var f:tab4):Integer,

która wyznaczy liczność najliczniejszego niezależnego podzbioru zbioru {1,...,n} wyznaczonego przez funkcję f.

Zad 16 (*) Dla danego typu tab01: array[1..n] of 0..1 napisz funkcję

maxrot(Const A:tab01):Integer,

która wyznaczy takie k, żeby jedynki i zera w tablicy powstałej przez przesunięcie A o k pozycji w prawo ułożyły się w możliwie dużą liczbę w zapisie dwójkowym. {Zrobienie tego w czasie liniowym jest możliwe, ale wykracza poza zakres materiału tego semestru}

Zad 17. Dane są dwa typy: cecha=array[0..n-1] of 0..1 oraz mantysa=array[0..m-1] of 0..1. Napisz funkcję

wieksza(const c1,c2:cecha; const m1,m2:mantysa):Boolean

która sprawdzi, czy liczba zapisana w systemie zmiennopozycyjnym o cesze c1 i mantysie m1 jest większa niż liczba o cesze c2 i mantysie m2. Zakładamy, że minimalna cecha reprezentuje 0 i że nie ma ukrytego bitu jednej drugiej.

Zad 18. Dla typów, jak w zadaniu poprzednim, wygeneruj kolejną co do wielkości liczbę rzeczywistą do danej.

Procedure kolejna(var c:cecha;var m:mantysa);

Jeśli mieliśmy największą możliwą liczbę w tablicach c:cecha, m:mantysa, to zostaw tablice w spokoju i niczego nie zmieniaj.

Procedure kolejna(var c:cecha;var m:mantysa);

Zad 19. Oblicz błąd względny wykonania działania 2/7+3/11 w systemie zmiennopozycyjnym z wykładu.