

Verificarea și Validarea Sistemelor Soft Laborator 3

Enunt A

Se da un vector X cu n componente întregi. Sa se conceapa un subalgoritm corect care determina cea mai lunga secventa $x[i], x[i+1], \dots, x[i+p]$ care are toate elementele distincte intre ele.

Completarea enunțului

Vectorul X are maximum 100 de elemente, iar elementele sunt numere intregi intre (inclusiv) -30 000 si 30 000.

Specificare

MaxDistincte(X , rezultat)

- Date de intrare:
 - X : o secvență de numere intregi,
 - prima.Lungime: numar natural egal cu numarul total de elemente din secventa X .
- Date de ieșire:
 - rezultat: un sir de numere intregi egal cu $(X_i, X_{i+1}, \dots, X_{i+n})$ unde toate elementele sunt distincte iar n este maxim ($0 \leq n \leq X.Lungime$, n natural).
 - Rezultat.Lungime: numar natural egal cu numarul total de elemente din sirul rezultat.

Date de test (cutia neagra)

Date invalide

Date de intrare: $X = (x)_{101}, x_i = 0, 0 \leq i < 101$

Date de intrare: $X = (x)_{100}, x_i = 0, 0 \leq i < 100, x_{93} = 30\ 001$

Date de intrare: $X = (x)_{100}, x_i = 0, 0 \leq i < 100, x_{15} = -30\ 001$

Date valide

Date de intrare: $X = \text{secventa vida}$

Date de iesire: secventa vida

Date de intrare: $X = (x)_{100}, x_i = 0$

Date de iesire: $(r)_1, r_i = 0$

Date de intrare: $X = (x)_{50}, x_i = i$

Date de iesire: $(r)_{50}, r_i = i$

Date de intrare: $X = (x)_{50}, x_i = i, 0 \leq i < 10; x_i = 0, 10 < i$

Date de iesire: $(r)_{10}, r_i = i$

Date de intrare: $X = (x)_{20}, x_i = 0, 0 \leq i < 10; x_i = i, 10 < i$

Date de iesire: $(r)_{10}, r_i = i + 10$

Date de intrare: $X = (x)_{25}, x_i = 0, 0 \leq i < 10 \text{ sau } 20 \leq i < 25; x_i = i - 10, 10 \leq i < 20$

Date de iesire: $(r)_{10}, r_i = i$

Verificarea și Validarea Sistemelor Soft Laborator 3

Date de intrare: $X = (x)_{30}$, $x_i = 0$, $0 \leq i < 10$ sau $20 \leq i < 25$; $x_i = i - 10$, $10 \leq i < 20$; $x_i = i - 25$, $25 \leq i < 30$
Date de iesire: $(r)_{10}$, $r_i = i$

Date de intrare: $X = (x)_{30}$, $x_i = 0$, $0 \leq i < 10$ sau $15 \leq i < 20$; $x_i = i - 10$, $10 \leq i < 15$; $x_i = i - 20$, $20 \leq i < 30$
Date de iesire: $(r)_{10}$, $r_i = i$

Date de test (cutia transparenta)

Date invalide

Date de intrare: $X = (x)_{101}$, $x_i = 0$, $0 \leq i < 101$

Date de intrare: $X = (x)_{100}$, $x_i = 0$, $0 \leq i < 100$, $x_{93} = 30\ 001$

Date de intrare: $X = (x)_{100}$, $x_i = 0$, $0 \leq i < 100$, $x_{15} = -30\ 001$

Date valide

Date de intrare: $X =$ secventa vida

Date de iesire: secventa vida

Date de intrare: $X = (x)_2$, $x_i = 0$

Date de iesire: $(r)_1$, $r_i = 0$

Date de intrare: $X = (x)_{10}$, $x_i = i$

Date de iesire: $(r)_{10}$, $r_i = i$

Date de intrare: $X = (x)_{30}$, $x_i = i$, $0 \leq i < 10$; $x_i = 0$, $10 < i$

Date de iesire: $(r)_{10}$, $r_i = i$

Date de intrare: $X = (x)_{20}$, $x_i = 0$, $0 \leq i < 10$; $x_i = i - 10$, $10 < i$

Date de iesire: $(r)_{10}$, $r_i = i$

Verificarea și Validarea Sistemelor Soft Laborator 3

Implementare

```
using System;
using System.Collections.Generic;
using System.Linq;
namespace L3
{
    public static class P11
    {
        public static IEnumerable<int> MaxDistincte(IEnumerable<int> sir)
        {
            if (sir == null)
                throw new ArgumentNullException("sir");

            if (sir.Count() > 100)
                throw new ArgumentException("Lungimea nu poate fi mai mare decat 100!", "sir");
            if (sir.Any(element => (element < -30000 || 30000 < element)))
                throw new ArgumentException("Elementele trebuie sa fie intregi si in intervalul [-30 000, 30 000]!");

            List<int> rezultat = new List<int>();
            ICollection<int> temp = new List<int>();

            foreach (int element in sir)
            {
                if (temp.Contains(element))
                {
                    if (temp.Count > rezultat.Count)
                    {
                        rezultat.Clear();
                        rezultat.AddRange(temp);
                    }
                    temp.Clear();
                }
                else
                    temp.Add(element);

                if (temp.Count > rezultat.Count)
                    return temp;
            }
            return rezultat;
        }
    }
}
```

Verificarea și Validarea Sistemelor Soft Laborator 3

Enunt B

Stabiliti pe criteriul cutiei transparente / negre, cazurile de test pentru urmatorul subalgoritm:

Subalgoritmul $\text{DetNrApMax}(n, X, k)$ este:
 $\{\phi: n \geq 1\}$

```
Fie k:=0; v:=0; p:=1; i:=1;
Cattimp i<=n execută
    Dacă  $x_i > x_p$  atunci
        p:=i; k:=1
    altfel
        Dacă  $x_p = x_i$  atunci
            k:=k+1
        sfdacă
            i:=i+1
    sfcat

    { $\psi: k = \text{nr.aparitiilor val.max}$ }
sf-DetApNrMax
```

Cutia neagra

Robustete

Date de intrare: $n = 0$

Date de intrare: $n = 1, X = (1)$

Date de iesire: $k = 1$

O singura aparitie a valorii maxime

Date de intrare: $n = 10, X = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$

Date de iesire: $k = 1$

Date de intrare: $n = 2, X = (10, 1)$

Date de iesire: $k = 1$

Date de intrare: $n = 5, X = (3, 10, 1, 1, 3)$

Date de iesire: $k = 1$

Date de intrare: $n = 3, X = (10, 1, 1)$

Date de iesire: $k = 1$

Date de intrare: $n = 3, X = (1, 1, 10)$

Date de iesire: $k = 1$

Mai multe aparitii ale valorii maxime

Date de intrare: $n = 5, X = (1, 2, 10, 10, 1)$

Date de iesire: $k = 2$

Date de intrare: $n = 5, X = (10, 10, 1, 2, 3)$

Date de iesire: $k = 2$

Verificarea și Validarea Sistemelor Soft Laborator 3

Date de intrare: $n = 5$, $X = (1, 2, 10, 10, 3)$

Date de iesire: $k = 2$

Date de intrare: $n = 5$, $X = (10, 10, 10, 10, 10)$

Date de iesire: $k = 5$

Cutia transparenta

Evitarea parcurgerii structurii cat timp:

Date de intrare: $n = 0$

Drumul: Cat timp \rightarrow altfel[1] \rightarrow atunci[2]

Date de intrare: $n = 1$, $X = (1)$

Drumul: Cat timp \rightarrow altfel[1] \rightarrow atunci[2] \rightarrow cat timp \rightarrow altfel[1]

Date de intrare: $n = 2$, $X = (2, 1)$

Drumul: Cat timp \rightarrow altfel[1] \rightarrow atunci[2] \rightarrow cat timp \rightarrow atunci[1]

Date de intrare: $n = 2$, $X = (2, 2)$

Drumul: Cat timp \rightarrow altfel[1] \rightarrow atunci[2] \rightarrow cat timp \rightarrow atunci[1]

Date de intrare: $n = 2$, $X = (1, 2)$