7/12/2023

Algorithm + Specification

B3 Task Munir Abood "Gambling"

spec ①

Specification: Input: $N[1...] \in \mathbb{N}$, $T[1....N] \in \mathbb{N}$

output: count $\in \mathbb{N}$

Precondition: –

Postcondition: $cnt = \sum_{A(N[i])}^{length(N)} 1$

Pattern: (Counting)

Algorithm:

Pattern	Task
length(x)	→ N
cnt	→ count
x[ ]	→ integers between 1 and N
A(x[i]	→ i → N

cnt = 0	
i = 1... length(x)	
A(x[i])	
T	F
cnt = cnt + 1	–

spec ②

Specification: Input: $N[1...] \in \mathbb{N}$,

Output: index $\in \mathbb{N}$, maxVal $\in \mathbb{N}$

Precondition: length(N) > 0

Postcondition: $1 \leq index \leq length(N)$ and $\forall i$ (1 ≤ i ≤ length(N)) : x[index] ≥ x[i] and maxVal = [ ] index].

$(maxIndex, MaxVal) = \max_{i=1}^{length(N)} (x[i])$

Pattern: (Maximum Selection)

Algorithm

Pattern	Task
length(x)	→ N
maxVal	→ maxcount
index	→ i
max(x[i])	→ max(count[i])

MaxIndex = 1	
MaxVal = 0	
i = 2... length(x)	
count > maxVal	
T	F
index = i [i]	
maxVal = count [i]	

Code:

```
using System;
using System.Collections.Generic;

namespace b3_code
{
    internal class Program
    {
        static void Main(string[] args)
        {
            string input = Console.ReadLine();
            string[] data = input.Split(' ');

            int N = int.Parse(data[0]);
            int M = int.Parse(data[1]);
            int[] T = new int[N];

            for (int i = 0; i < N; i++)
            {
                T[i] = Convert.ToInt32(Console.ReadLine());
            }

            int maxcount = 0;
            int count = 0;

            List<int> NonDuplicates = new List<int>();

            for (int i = 0; i < N; i++)
            {
                count = 0;

                for (int j = 0; j < N; j++)
                {
                    if (T[i] == T[j])
                    {
                        count++;
                    }
                }

                if (count > maxcount)
                {
                    maxcount = count;
                }
```

```
        }

        for (int i = 0; i < N; i++)
        {
            count = 0;

            for (int j = 0; j < N; j++)
            {
                if (T[i] == T[j])
                {
                    count++;
                }
            }

            if (count == maxcount && !NonDuplicates.Contains(T[i]))
            {
                Console.WriteLine($"{i + 1} {T[i]} {count}");
                NonDuplicates.Add(T[i]);
            }
        }
    }
  }
}
```