

Specification

Shivam Taneja, CompSci 2XB3 Project

April 11, 2019

This Module Interface Specification (MIS) document contains modules, types and methods for implementing part of 2XB3 Project - SafeParkingZone

Theft Distance Module

Module

TheftDis

Uses

Location

Syntax

Exported Constants

N/A

Exported Types

N/A

Exported Access Programs

Routine name	In	Out	Exceptions
result	String, \mathbb{N}	Seq of Location	IOException

Semantics

State Variables

TheftZones: Seq of Location

State Invariant

None

Access Routine Semantics

result(*data*, *threshold*):

- transition: $TheftZones := open(file) \Rightarrow (\forall element \in file : element.Dist < threshold \Rightarrow TheftZones = TheftZones + [element])$
- output: $out := TheftZones$

- exception: $\neg \text{open}(\text{file}) \Rightarrow \text{IOException}$

The Module

Theft Frequency Module

TheftFreq

Uses

Location
TheftDis

Syntax

Exported Types

None

Exported Constants

None

Exported Access Programs

Routine name	In	Out	Exceptions
freqDis	2D list of Locations	Seq of Integer	None
freqNor	Seq of Location	N	None
dist	Seq of Location	2D list of Location	None

Semantics

State Variables

TheftFreq: Seq of Integer
TheftList: 2D list of Locations

State Invariant

$| \textit{TheftList} | = 10$

Access Routine Semantics

freqDis(*input*):

- transition: $TheftFreq := \forall i \in [0..9] : TheftFreq[i] = | input[i] |$
- output: $out := TheftFreq$

freqNor(*in*):

- transition: $TheftFreq := | in |$
- output: $out := TheftFreq$

dist(*input*, *threshold*):

- transition: $TheftList :=$
 $\forall i \in [0..9] : (\forall j \in [0.. | input | - 1] : input[j].dist < (threshold/10) * (i + 1) \Rightarrow$
 $TheftList[i] = TheftList[i] + input[j])$
- output: $out := TheftList$

Search Algorithm Module

Template Module

SearchAlg

Uses

None

Syntax

Exported Access Programs

Routine name	In	Out	Exceptions
search	Seq of Character (Text), Seq of Character(Pattern)	\mathbb{N}	None

Semantics

State Variables

None

State Invariant

$NO_OF_CHARS = 256$

Access Routine Semantics

$search(txt, pat):$

- $out := m \leftarrow |pat| \wedge n \leftarrow |txt| \wedge s = 0 \Rightarrow (badchar[NO_OF_CHARS] \Rightarrow badChar(pat, |pat|, badchar) \Rightarrow While(s \leq n - m) do j \leftarrow m \wedge While(j > 0 \wedge pat[j] = txt[s + j] do j = j - 1 \Rightarrow (j < 0)?out \rightarrow true : s = s + max(1, j - badchar[txt[s + j]])))$

Local Functions

$max : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$

$$\max(a, b) \equiv (a > b \Rightarrow a) \vee (a < b \Rightarrow b)$$

$\text{badChar} : \text{Seq of character} \times \mathbb{N} \rightarrow \text{Seq of Integer}$

$$\begin{aligned} \text{badChar}(str, size, badchar) &\equiv \forall i \in [0..NO_OF_CHARS] - 1 : badchar[i] = -1 \Rightarrow \\ \forall i \in [0..size - 1] : badchar[str[i]] &= i \end{aligned}$$