

ALGORITHMEN ZUM IRRGARTEN-PROBLEM

ALGORITHMUS	<i>SucheWegInIrrgarten</i>	
I/O-OBJEKTE	Irrgarten:	Array[1..n,1..m] of Char
	WegGefunden:	Boolean
Inp-OBJEKT	x, y:	Integer
<ul style="list-style-type: none"> • Falls Irrgarten[x,y] = ‚Z‘ <ul style="list-style-type: none"> dann • WegGefunden ← true sonst • Falls (Irrgarten[x,y] = ‚S‘) oder (Irrgarten[x,y] = ‚ ‚) <ul style="list-style-type: none"> dann • Irrgarten[x,y] ← ‚O‘ <ul style="list-style-type: none"> • SucheWegInIrrgarten(Irrgarten, WegGefunden, x+1, y) • Falls nicht WegGefunden <ul style="list-style-type: none"> dann • SucheWegInIrrgarten(Irrgarten, WegGefunden, x, y-1) • Falls nicht WegGefunden <ul style="list-style-type: none"> dann • SucheWegInIrrgarten(Irrgarten, WegGefunden, x-1, y) • Falls nicht WegGefunden <ul style="list-style-type: none"> dann • SucheWegInIrrgarten(Irrgarten, WegGefunden, x, y+1) • Falls nicht WegGefunden <ul style="list-style-type: none"> dann • Irrgarten[x,y] ← ‚ ‚ 		

Aufruf:

- Lies_Irrgarten_aus_Datei(Irrgarten)
- Gefunden ← false
- SucheWegInIrrgarten(Irrgarten, Gefunden, 7, 5)

ALGORITHMUS	<i>SucheWegInIrrgarten_Kürzester Weg</i>	
I/O-OBJEKTE	minIrrgarten:	Array[1..n,1..m] of Char { <i>gesuchte Lösung</i> }
	minLaenge:	Integer { <i>Länge des kürzesten Weges</i> }
Inp-OBJEKT	x, y:	Integer
	aktIrrgarten:	Array[1..n,1..m] of Char { <i>aktuelle Lösung</i> }
	aktLaenge:	Integer { <i>Länge des aktuellen Weges</i> }
<ul style="list-style-type: none"> • Falls aktLaenge < minLaenge { <i>sonst braucht nicht mehr weiter gesucht zu werden</i> } <ul style="list-style-type: none"> dann • Falls Irrgarten[x,y] = ‚Z‘ <ul style="list-style-type: none"> dann • minLaenge ← aktLaenge • minIrrgarten ← aktIrrgarten sonst • Falls (Irrgarten[x,y] = ‚S‘) oder (Irrgarten[x,y] = ‚ ‚) <ul style="list-style-type: none"> dann • aktIrrgarten[x,y] ← ‚O‘ <ul style="list-style-type: none"> • SucheWegInIrrgarten_KürzesterWeg (minIrrgarten, minLaenge, x+1, y, aktIrrgarten, aktLaenge+1) • SucheWegInIrrgarten_KürzesterWeg (minIrrgarten, minLaenge, x, y-1, aktIrrgarten, aktLaenge+1) • SucheWegInIrrgarten_KürzesterWeg (minIrrgarten, minLaenge, x-1, y, aktIrrgarten, aktLaenge+1) • SucheWegInIrrgarten_KürzesterWeg (minIrrgarten, minLaenge, x, y+1, aktIrrgarten, aktLaenge+1) 		

Aufruf:

- Lies_Irrgarten_aus_Datei(Irrgarten1)
- Init_Irrgarten(Irrgarten2)
- minLaenge ← MaxInt { $2^{15} - 1$ }
- SucheWegInIrrgarten_KürzesterWeg(Irrgarten2, minLaenge, 7, 5, Irrgarten1, 1)