

## פתור לי פאזל

בחידה של החודש הקודם התבקשתם לכתוב קוד הפותר פאזל, כאשר הקלט הוא תמונה המחולקת ל-  $n^2$  מלבנים.



## פתרון

הפתרון במקרה זה מאוד פשוט. הקלט של האלגוריתם הוא התמונה, ריבוע המורכב מאוסף של ריבועים קטנים. מה שהאלגוריתם עושה זה לחתוך את התמונה המקורית ל-  $n^2$  חלקים, ולהשוות בין החלקים שהאלגוריתם חישב לחלקים שהוא קיבל כקלט. כל חלק מזווג לחלק אחד ויחיד - וכך הפאזל נפתר. מצורף את הקוד שפותר את הבעייה.

```
(* Here we are defining a function called mysolvepazel that accepts two arguments: myImage (an image) and myparts (a list of images). *)
```

```
mysolvepazel[myImage_Image,myparts:{{_Image}}]:=
```

```
(* Module is a scoping construct that allows the creation and manipulation of local variables. *)
```

```
Module[{d1,d2,theCorectparts,i,j,k,ans={},p1},
```

```
(* We are capturing the dimensions of the image myImage into d1. *)
```

```
d1=ImageDimensions[myImage];
```

```
(* We are capturing the dimension of the first image in the list myparts into d2. We are considering only the first element of the resulting list. *)
```

```
d2=ImageDimensions[myparts[[1]][[1]]];
```

```
(* Here we are partitioning the original image into smaller images with the dimensions specified by d2. This gives us a list of images that should match the ones in myparts. *)
```

```
theCorectparts=ImagePartition[myImage,d2];
```

```
(* We're running a For loop that iterates over the length of myparts. *)
```

```
For[k=1,k<=Length[myparts],k++,
```

```
(* We are creating a boolean matrix which contains True at the positions where the corresponding images in theCorectparts and myparts are the same. *)
```

```
p1=Position[Table[theCorectparts[[i,j]]===myparts[[k]],{i,1,Length[theCorectparts]},{j,1,Length[theCorectparts[[i]]}],True];
```

```
(* Appending the positions where images match to the list ans. *)
```

```
AppendTo[ans,p1];
```

```
];
```

```
(* The function returns the list ans, which contains the positions of the images in myparts that are found in theCorectparts. *)
```

```
ans
```

```
];
```

במילים אחרות, הפונקציה הזו מחפשת את המיקומים של תמונות קטנות (myparts) בתוך תמונה גדולה יותר (myImage). היא עושה זאת על ידי חלוקת התמונה הגדולה לבלוקים באותו הגודל של התמונות הקטנות, ואז משווה את הבלוקים הללו לתמונות הקטנות. הפלט הוא רשימת מיקומים שבהם כל תמונה קטנה נמצאה בתוך התמונה הגדולה.