Spring 2018 SIT22004

ICT Problem Solving

Discussion: PA #5

April 11, 2018

PA5: Polyomino Puzzle

A polyomino is a rectilinear shape obtained by combining one or more 1x1 square blocks. For instance, Figure 1 shows four polyominos with different combinations of such blocks. Suppose that we are playing with a puzzle to form a square with given polyominos. In this puzzle, each polyomino is not allowed to rotate, and all polyominos are required to be used to form a square. Figure 2 shows how we can achieve a square by arranging the four polyominos in Figure 1.

Write a program that find a solution of this puzzle for given polyominos.

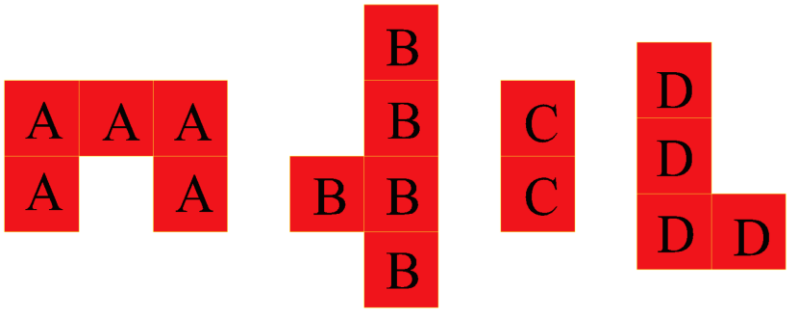


Figure 1. Four polyominos

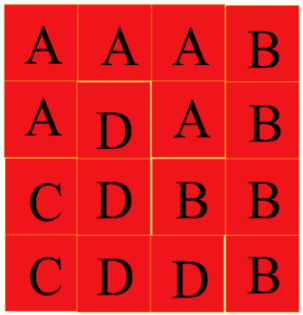


Figure 2. A square composed of the four polyominos in Figure 1.

(Continued)

Requirements

Input Data

- The first line from the standard input has an integer n, the number of polyominos in this puzzle for $1 \leq n \leq 5$. The following lines define the shapes of n polyominos. These polyominos in sequence have IDs from 1 to n, respectively.
- For each polyomino n, $1 \leq n \leq 5$, the first line gives two numbers h and w which represent the height and the width of a polyominos respectively, for $1 \leq h \leq 4$ and $1 \leq w \leq 4$. Then, h lines follow, each of which contains a number consisting of w binary digits, which defines the shape of a polyomino. A binary digit is 1 if and only if the polyomino has a square block at the corresponding position.

Output Data

- Print out an array that represents the arrangement of the given polyominos to the standard output. The array should be a square, that is, the number of rows should be the same as that of the columns. A square block in the array should be specified by the ID of the polyomino at the corresponding position. If it is impossible to form a square array, then print "No solution possible" as the output.
- Your program must return the result within 3.0 seconds

(Continued)

Example of test data

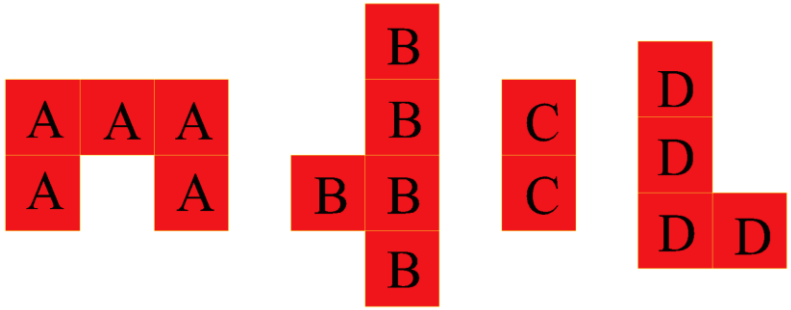
Input data

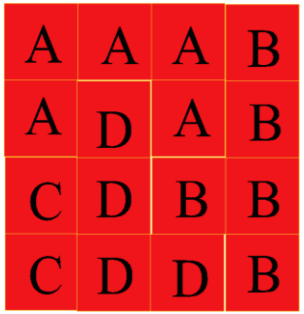
```
4
2 3
1 1 1
1 0 1
4 2
0 1
0 1
1 1
0 1
2 1
1
1
3 2
1 0
1 0
1 1
```

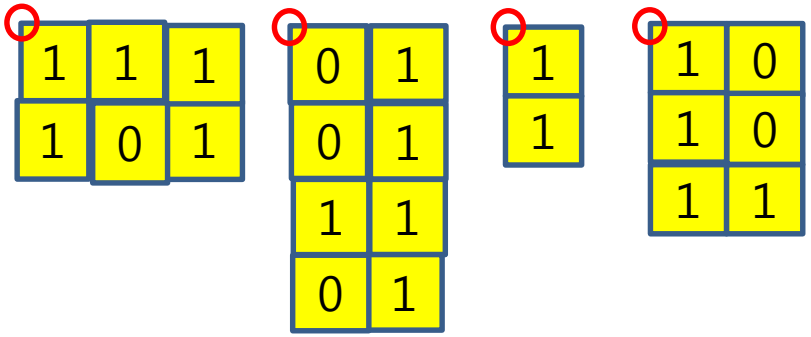
Output data

```
1 1 1 2
1 4 1 2
3 4 2 2
3 4 4 2
```

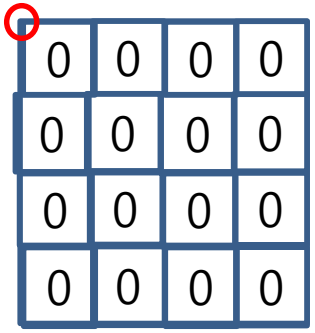
How to model polyominos





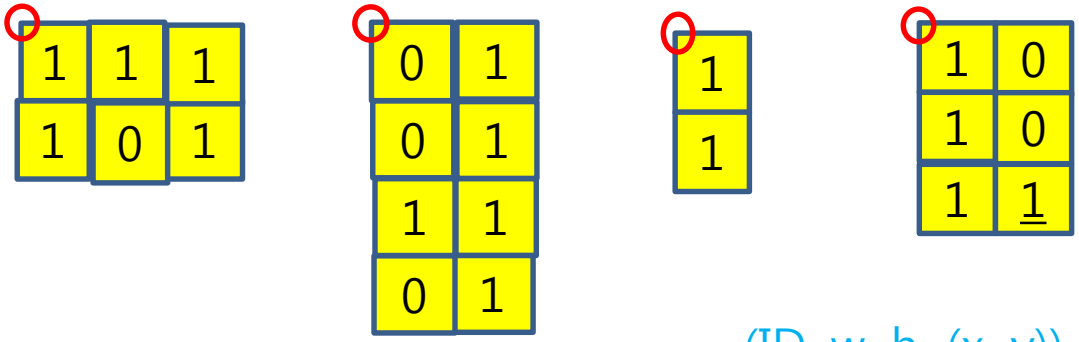


(ID w,h, (x, y))

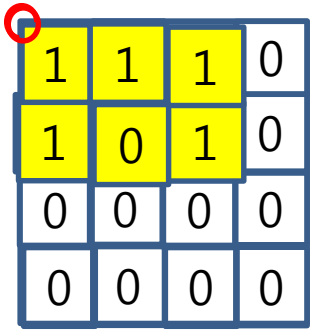


How to specify the square board?

How to place polyminos

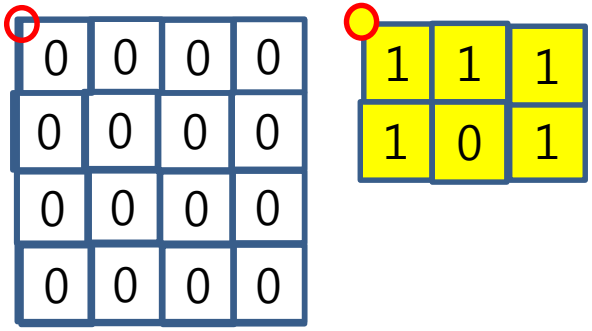


(ID, w, h, (x, y))

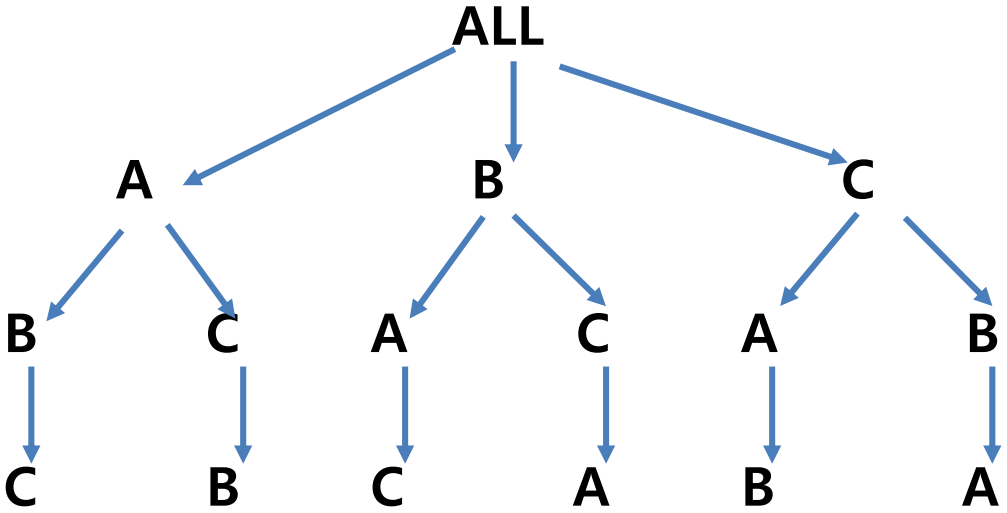


Basic Strategy

Backtracking



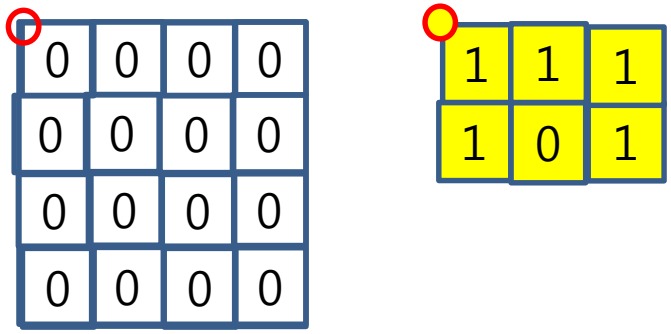
depth-first-search



(ID, w, h, (x, y))

How to check a right placement

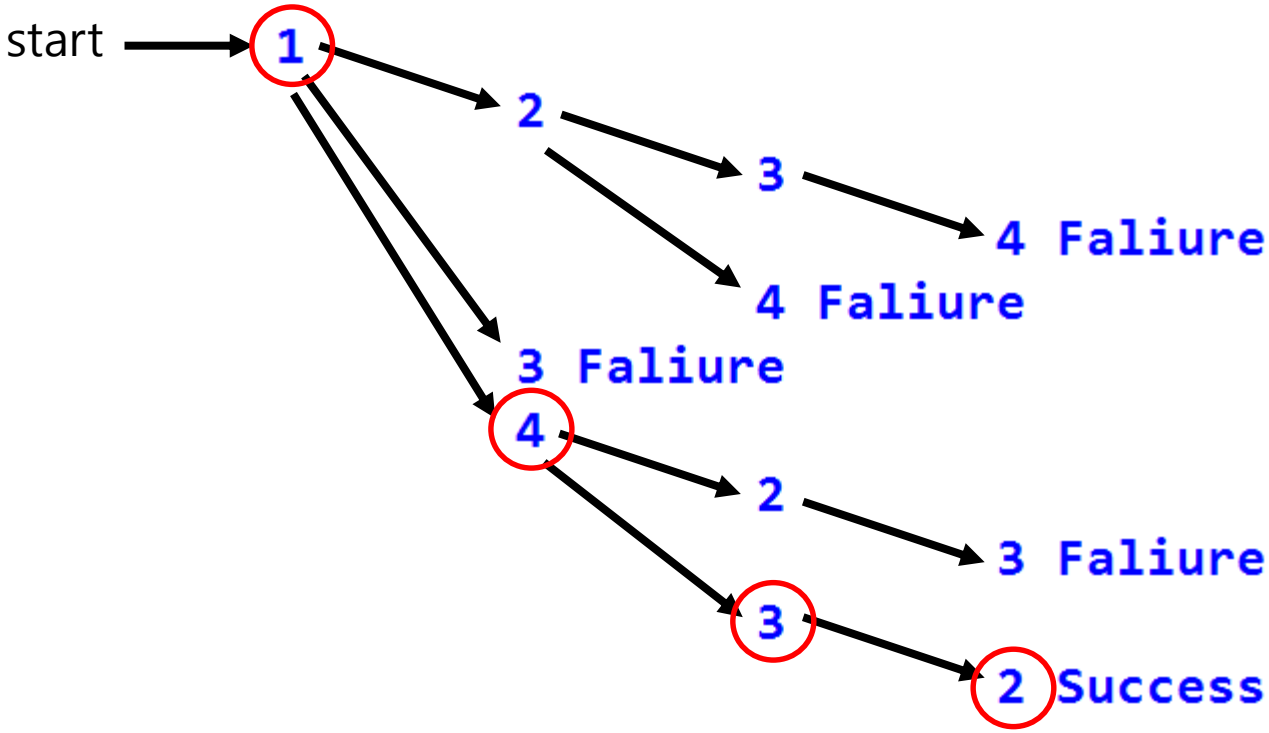
For each block of a polyomino with a value of 1, check if the corresponding block has a value of 0.



Also check if the polyomino is completely contained in the square-board.

How to permute list items

- Given a list, find the sequence of its items that can be fitted to a board.



A given list:
[1,2,3,4]

- Example code

```
from random import randint

def fitted(m):
    return randint(0,1) == 0

def get_msg(is_fitted,is_empty):
    if is_fitted:
        if is_empty:
            return "Success"
        return ""
    return "Faliure"
```

```python
def permutation(lst, success = False, level = 0):
    for i in range(len(lst)):
        m, remLst = lst[i], lst[:i] + lst[i+1:]
        is_empty = len(remLst) == 0
        is_fitted = fitted(m)

        print("\t" * level, m, get_msg(is_fitted, is_empty))
        if is_fitted:
            success, level = permutation(remLst, is_empty, level+1)
            if success: break

    return success, level-1
```

```python
data = [1,2,3,4]
if permutation(data)[0]:
    print("\n Success")
else:
    print("\n Failue")
```

* Expected execution results

```
1 Faliure
2 Faliure
3
        1
                2 Faliure
                4
                        2 Faliure
        2
                1
                        4 Faliure
                4
                        1 Success

Success
```

```
1
        2
                3 Faliure
                4
                        3 Success

Success
```