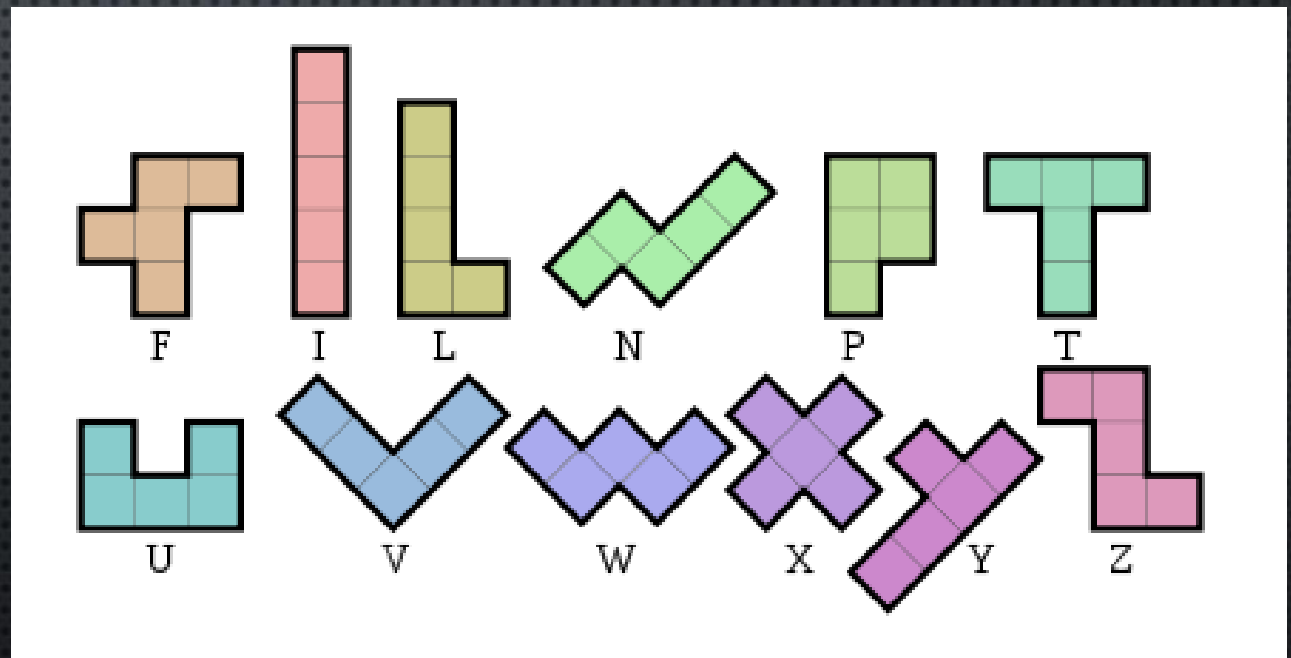


**METODE ZA  
RJEŠAVANJE LOGIČKE  
ZAGONETKE  
PENTOMINO**

# PENTOMINA?

- GEOMETRIJSKI OBLICI STVORENI SPAJANJEM 5 KVADRATA U JEDNU CJELINU
- FORMIRAJU OBLIKE KOJI PODSJEĆAJU NA SLOVA I KORISTE SE U RAZLIČITIM LOGIČKIM ZADACIMA



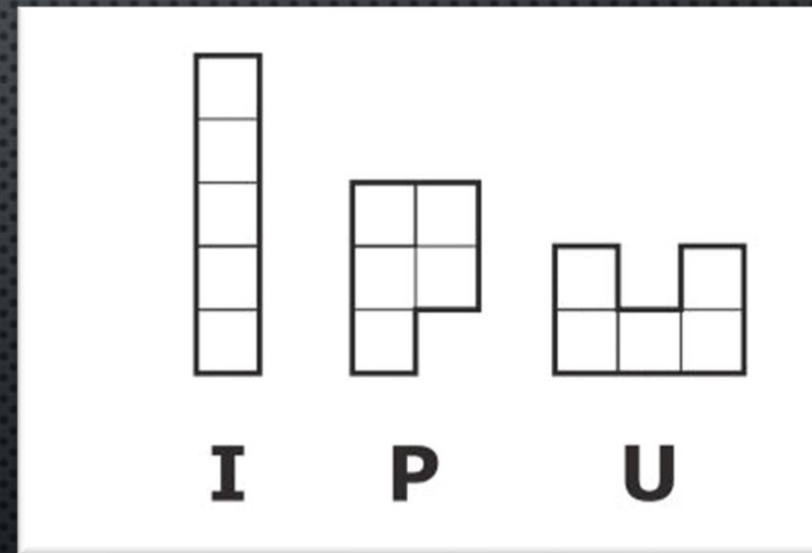
# PENTOMINO



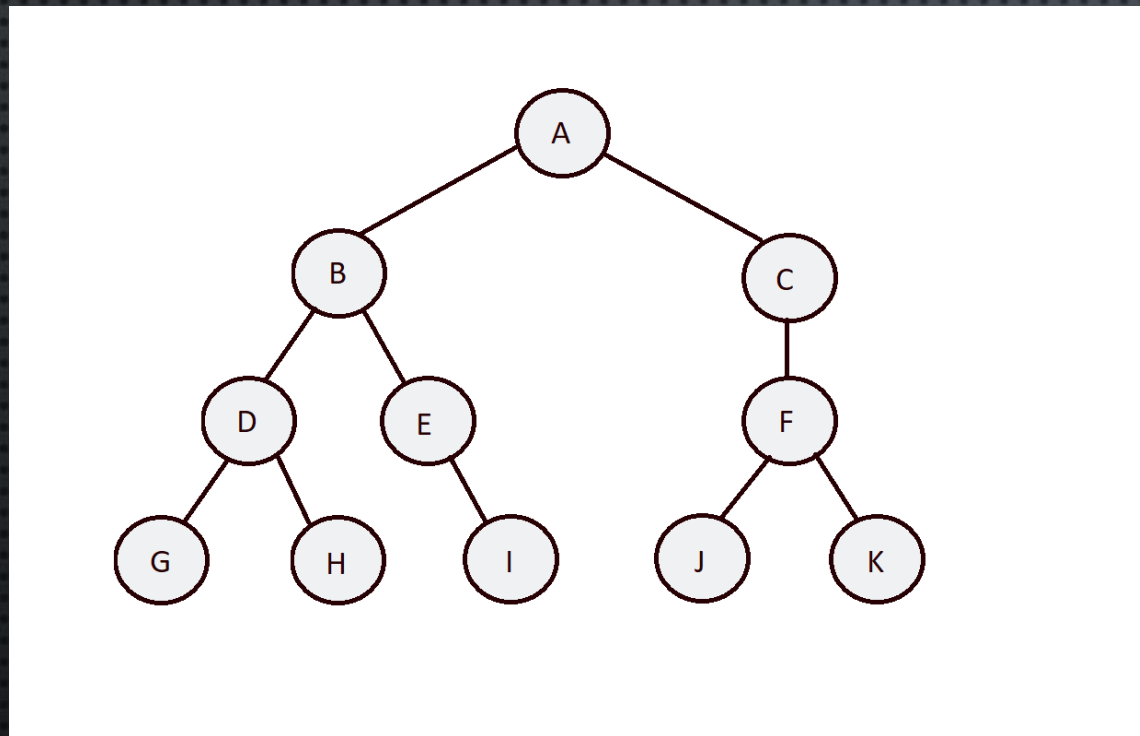
- TABLICA DIMENZIJA 13x10 U KOJU TREBA UNIJETI SVIH 12 PENTOMINA
- PENTOMINA SE NE SMIJU DODIRIVATI ČAK NI DIJAGONALNO
- SVAKI ZADATAK IMA NEKOLIKO DEFINIRANIH PRAZNIH POLJA, KOJI OMOGUĆUJU JEDINSTVENOST RJEŠENJA

# PRIMJER MINI ZAGONETKE

	4			3	
	U	U		P	P
	U			P	P
	U	U		P	
∅					
	I	I	I	I	I

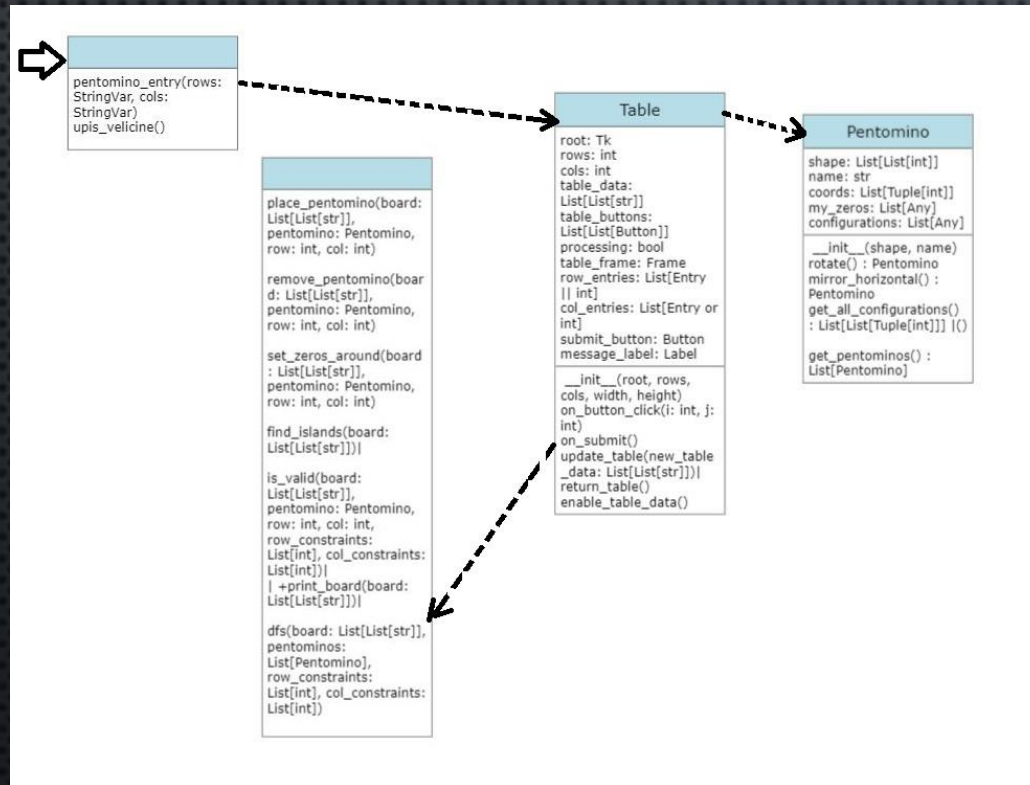


# DFS (DEPTH FIRST SEARCH) ALGORITAM



- PROPAGACIJA OGRANIČENJA
- REZANJE GRANA KOJE SIGURNO NE VODE DO RJEŠENJA

# PENTOMINO.PY, DFS\_CP.PY, ZAGONETKA.PY I UPIS\_DIMENZIJA.PY





```

pentomino_entry(rows:
StringVar, cols:
StringVar)
upis_velicine()

```

```

place_pentomino(board:
List[List[str]],
pentomino: Pentomino,
row: int, col: int)

remove_pentomino(board:
List[List[str]],
pentomino: Pentomino,
row: int, col: int)

set_zeros_around(board:
List[List[str]],
pentomino: Pentomino,
row: int, col: int)

find_islands(board:
List[List[str]])

is_valid(board:
List[List[str]],
pentomino: Pentomino,
row: int, col: int,
row_constraints:
List[int], col_constraints:
List[int])
| +print_board(board:
List[List[str]])

dfs(board: List[List[str]],
pentominos:
List[Pentomino],
row_constraints:
List[int], col_constraints:
List[int])

```

```

Table
root: Tk
rows: int
cols: int
table_data:
List[List[str]]
table_buttons:
List[List[Button]]
processing: bool
table_frame: Frame
row_entries: List[Entry
|| int]
col_entries: List[Entry or
int]
submit_button: Button
message_label: Label

__init__(root, rows,
cols, width, height)
on_button_click(i: int, j:
int)
on_submit()
update_table(new_table
_data: List[List[str]])
return_table()
enable_table_data()

```

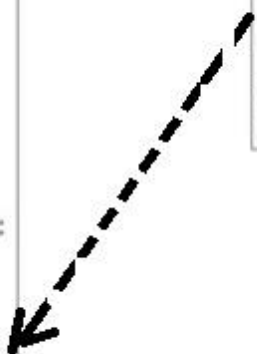
```

Pentomino
shape: List[List[int]]
name: str
coords: List[Tuple[int]]
my_zeros: List[Any]
configurations: List[Any]

__init__(shape, name)
rotate() : Pentomino
mirror_horizontal() :
Pentomino
get_all_configurations()
: List[List[Tuple[int]]] |()

get_pentominos() :
List[Pentomino]

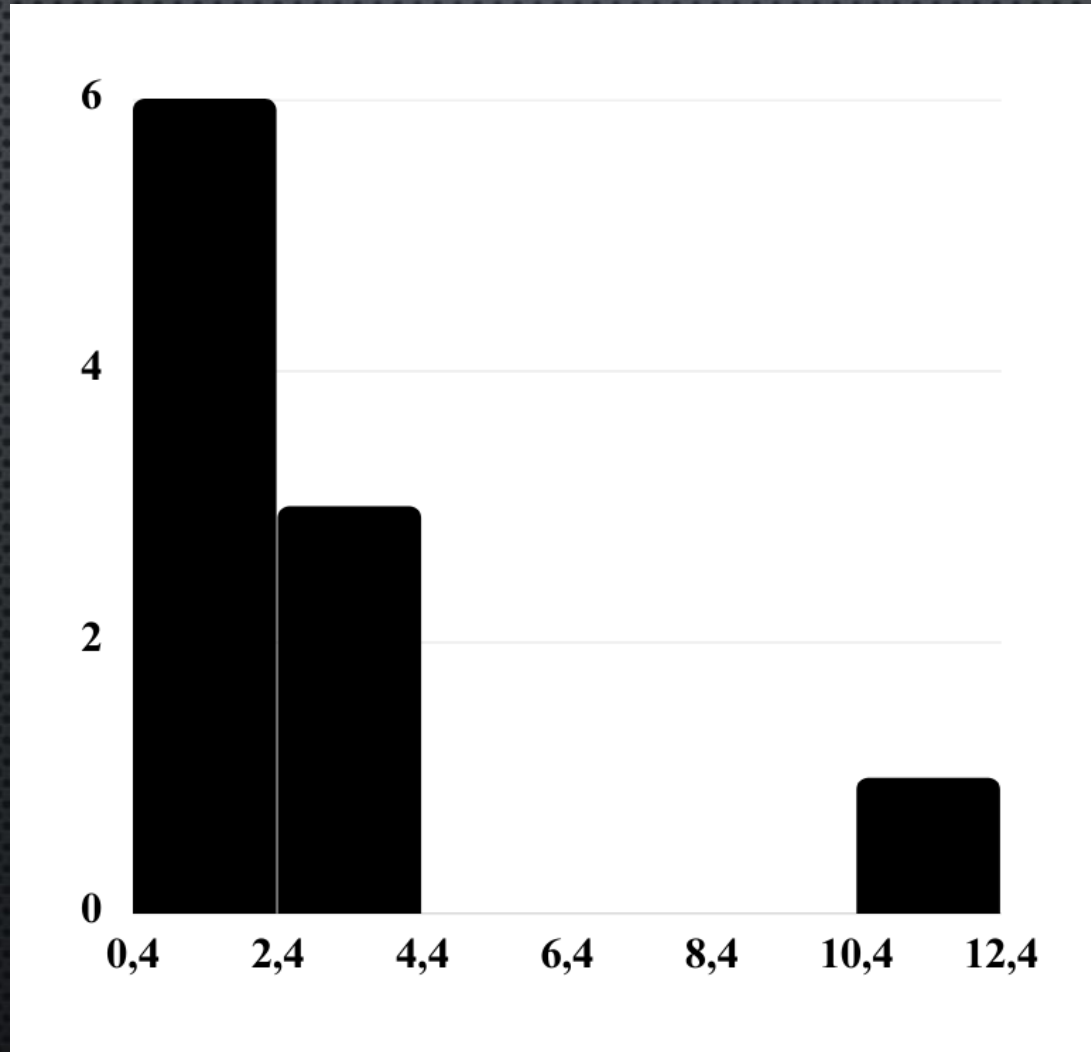
```



# VRIJEME POTREBNO ZA DOLAZAK DO RJEŠENJA

Redni broj	1	2	3	4	5	6	7	8	9	10	Prosjek
Vrijeme rješavanja [s]	3.6	1.7	11.2	1.6	1.7	3.9	4.1	1.3	0.4	0.5	3





HVALA NA PAŽNJI

MONIKA STRACENSKI