

Filimon Adrian  
334 CC

③ Sudoku

Soluție iterativă: backtracking.

Soluție paralelă:

~~Sol~~

Soluția poate fi găsită în alți pași:

① BFS de la începutul tablei  $\rightarrow$  primirea  $\times$  poziției  
de început

② Reducerea ficării soluției din cele  $\times$  găsite.

funcț solve(sol, explored, grid)

```
while (sol  $\neq$   $\emptyset$ ) {
```

```
    waiting = thr ++;
```

```
    thr.wait();
```

```
    waiting = thr --;
```

```
}
```

```
aux = sol.pop();
```

```
index, value = MIN-VAL(grid.unsolved);
```

```
Reduce(index, value, grid)
```

```
if (Valid-grid(grid)) {
```

```
    sol.push(grid-i);
```

```
    explored.push(grid-i);
```

```
}
```

```
thr.notify();
```

```

else {
    explored.push(guid);
}

return guid - i;

```

R Apoi, o să am nevoie de un pool de threads,

pentru a lua fiecare posibilitate și a o rezolva

```
func thread-pool(nr-thr, guid)
```

```
for (nr-thr) {
```

```
    spawn-thread(solve(sol, explored, guid))
```

```
}
```

```
}
```