E-fólio A | Instruções para a realização do E-fólio

INTRODUCTION TO ARTIFICIAL INTELLIGENCE | 21071 | 2023/2024

Folio A is inspired by a problem of the placement of police stations, sheriffs and deputies in the Old West.

As settlers established themselves throughout the territory and exploited the land with various productive activities, they also created the conditions for criminal activity. It became clear that some kind of protection was needed so that the settlers could work without worry. Thus, for each territory, a monthly sum was set aside for security. With this money, the governor could maintain police stations, with their respective sheriffs and deputies.

To better protect the population, the governor of a territory collected the distribution of the population by zones, dividing the territory into NxM equal parts, and indicating for each zone how many families had settled.

4		8	4					
				3			1	
	3							
			1	1			3	
		2	4				1	
	2			8		4	3	10
		3			4			

There are some areas without any families, but there are areas with 10 families, as shown in the picture opposite.

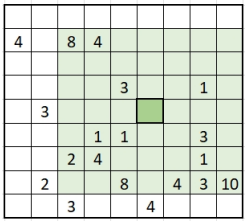
We now want to position the police stations (each station has a sheriff) and the deputies. Both the sheriff and the deputies are on horseback, so there's no problem because the roads are poor, but they do have a limit to the range they can travel within.

speed. Thus, a police station can only protect the zone in which it is placed, and a group of zones nearby, depending on the number of deputies:

MPs	Protection radius
0 (only the sheriff)	1 (protects 3x3 zones)
1	2 (protects 5x5 zones)
5	3 (protects 7x7 zones)
13	4 (protects 9x9 zones)

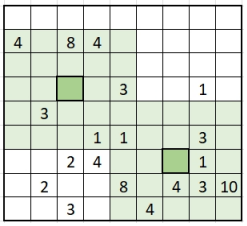
As funds are limited, you have to choose whether to have 1 or more police stations, and how many deputies to put in each station, in order to be able to protect a minimum number of families. A police station and its sheriff cost 4 gold coins. For each deputy you need 1 gold coin.

In the example map, the amount available is 12 gold coins and at least 57 families need to be protected.



One possibility would be to make a single precinct and put 5 deputies there. As 1 police station costs 4, and each deputy costs 1, the total cost will be 4+5=9, leaving 3 coins unused. The police station protects a radius of 3, leaving 7x7 protected zones. In this attempt, the police station has been placed in a position marked in dark green. In light green are the protected zones.

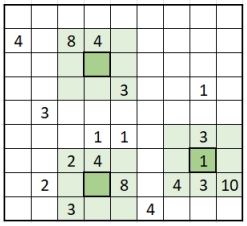
Unfortunately, this police station only protects 53 families, not the 57 that are needed. So this is not a valid solution.



Another possibility is to use two police stations, with a radius of 2, costing: 4+4+1+1=10, leaving 2 coins unused.

Police stations in these positions protect 57 families, which is a valid solution.

Finally, we can still have 3 police stations without deputies, since each police station costs 4, spending the entire budget of 12 coins on police stations.



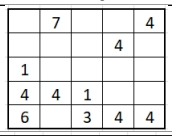
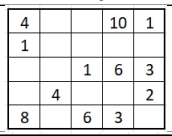
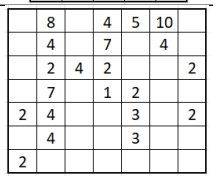
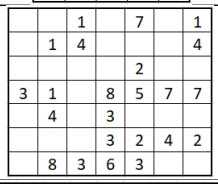
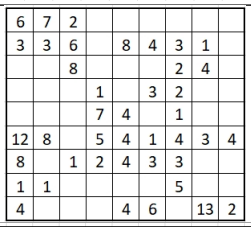
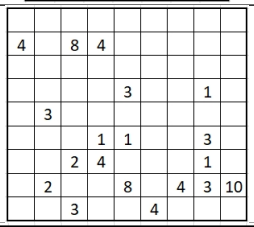
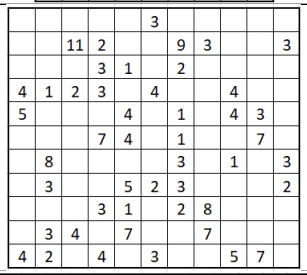
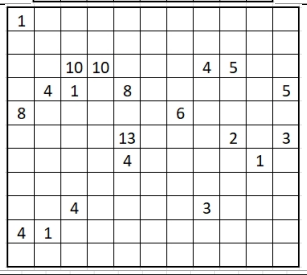
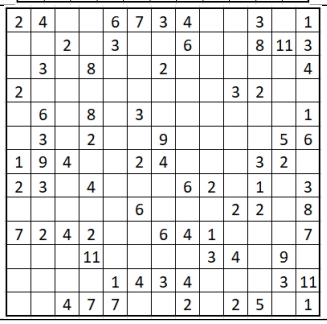
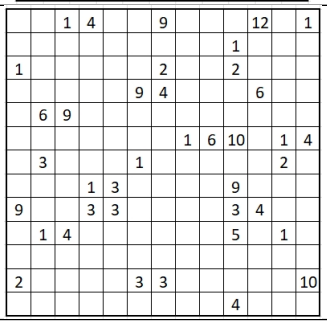
Unfortunately, this set of three police stations only protects 53 families, and not the 57 needed, so it's not a valid solution.

Consider the following parameters of 10 instances:

ID	N	M	Money	ProtectA	ProtectB
1	5	5	4	19	20
2	5	5	4	21	22
3	7	7	8	67	68
4	7	7	8	59	60
5	9	9	12	125	126
6	9	9	12	57	58
7	11	11	16	140	141
8	11	11	16	93	94
9	13	13	20	211	212
10	13	13	20	125	126

Each instance has two versions, version A ($_{ProtectA}$) and version B ($_{ProtectB}$). Everything else is the same, only the number of families the governor wants to protect changes.

In addition to the parameters, the distribution of families among the zones is defined for each instance:

ID	Map	ID	Map
1		**2**	
3		**4**	
5		**6**	
7		**8**	
9		**10**	

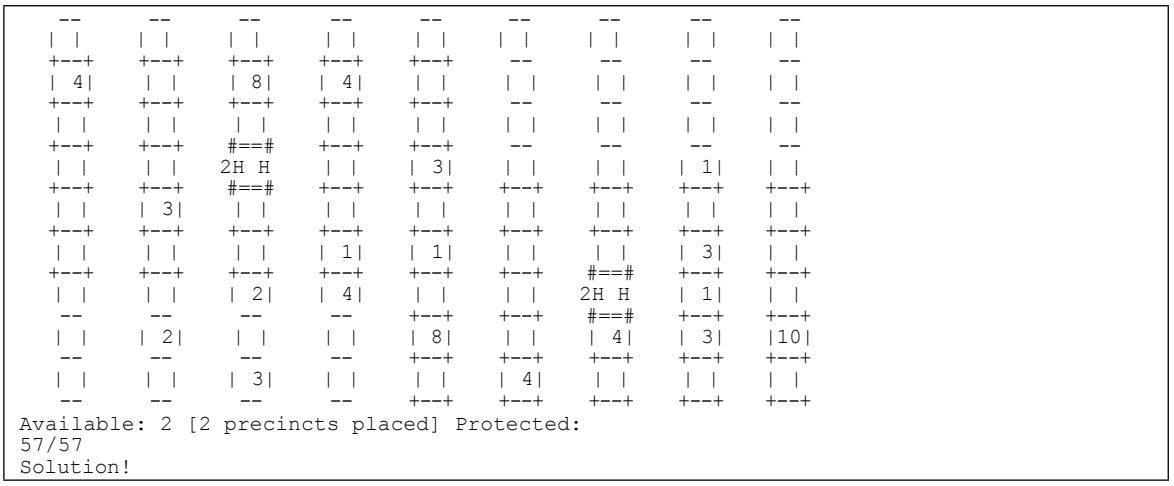
Instance maps can be defined as matrices with a maximum value of 20x20, and be initialized statically in the code:

```
{
                {// 5x5
{0,7,0,0,4},
{0,0,0,4,0},
{1,0,0,0,0},
{4,4,1,0,0},
{6,0,3,4,4},
},
{
{4,0,0,10,1},
{1,0,0,0,0},
{0,0,1,6,3},
{0,4,0,0,2},
{8,0,6,3,0},
},
                {// 7x7
{0,8,0,4,5,10,0},
{0,4,0,7,0,4,0},
{0,2,4,2,0,0,2},
{0,7,0,1,2,0,0},
```

```
{2,4,0,0,3,0,2},
{0,4,0,0,3,0,0},
{2,0,0,0,0,0,0},
},
{
{0,0,1,0,7,0,1},
{0,1,4,0,0,0,4},
{0,0,0,0,2,0,0},
{3,1,0,8,5,7,7},
{0,4,0,3,0,0,0},
{0,0,0,3,2,4,2},
{0,8,3,6,3,0,0},
},
                {// 9x9
{6,7,2,0,0,0,0,0,0},
{3,3,6,0,8,4,3,1,0},
{0,0,8,0,0,0,2,4,0},
{0,0,0,1,0,3,2,0,0},
{0,0,0,7,4,0,1,0,0},
{12,8,0,5,4,1,4,3,4},
{8,0,1,2,4,3,3,0,0},
{1,1,0,0,0,0,5,0,0},
{4,0,0,0,4,6,0,13,2},
},
{
{0,0,0,0,0,0,0,0,0},
{4,0,8,4,0,0,0,0,0},
{0,0,0,0,0,0,0,0,0},
{0,0,0,0,3,0,0,1,0},
{0,3,0,0,0,0,0,0,0},
{0,0,0,1,1,0,0,3,0},
{0,0,2,4,0,0,0,1,0},
{0,2,0,0,8,0,4,3,10},
{0,0,3,0,0,4,0,0,0},
},
                {// 11x11
{0,0,0,0,0,3,0,0,0,0,0},
{0,0,11,2,0,0,9,3,0,0,3},
{0,0,0,3,1,0,2,0,0,0,0},
{4,1,2,3,0,4,0,0,4,0,0},
{5,0,0,0,4,0,1,0,4,3,0},
{0,0,0,7,4,0,1,0,0,7,0},
{0,8,0,0,0,0,3,0,1,0,3},
{0,3,0,0,5,2,3,0,0,0,2},
{0,0,0,3,1,0,2,8,0,0,0},
{0,3,4,0,7,0,0,7,0,0,0},
{4,2,0,4,0,3,0,0,5,7,0},
},
{
{1,0,0,0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,0},
{0,0,10,10,0,0,0,4,5,0,0},
{0,4,1,0,8,0,0,0,0,0,5},
{8,0,0,0,0,0,6,0,0,0,0},
{0,0,0,0,13,0,0,0,2,0,3},
{0,0,0,0,4,0,0,0,0,1,0},
{0,0,0,0,0,0,0,0,0,0,0},
{0,0,4,0,0,0,0,3,0,0,0},
{4,1,0,0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,0},
},
                {// 13x13
{2,4,0,0,6,7,3,4,0,0,3,0,1},
{0,0,2,0,3,0,0,6,0,0,8,11,3},
{0,3,0,8,0,0,2,0,0,0,0,0,4},
{2,0,0,0,0,0,0,0,0,3,2,0,0},
{0,6,0,8,0,3,0,0,0,0,0,0,1},
{0,3,0,2,0,0,9,0,0,0,0,5,6},
{1,9,4,0,0,2,4,0,0,0,3,2,0},
{2,3,0,4,0,0,0,6,2,0,1,0,3},
{0,0,0,0,0,6,0,0,0,2,2,0,8},
{7,2,4,2,0,0,6,4,1,0,0,0,7},
{0,0,0,11,0,0,0,0,3,4,0,9,0},
{0,0,0,0,1,4,3,4,0,0,0,3,11},
{0,0,4,7,7,0,0,2,0,2,5,0,1},
},
{
{0,0,1,4,0,0,9,0,0,0,12,0,1},
{0,0,0,0,0,0,0,0,0,1,0,0,0},
{1,0,0,0,0,0,2,0,0,2,0,0,0},
{0,0,0,0,0,9,4,0,0,0,6,0,0},
{0,6,9,0,0,0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,1,6,10,0,1,4},
{0,3,0,0,0,1,0,0,0,0,0,2,0},
{0,0,0,1,3,0,0,0,0,9,0,0,0},
{9,0,0,3,3,0,0,0,0,3,4,0,0},
{0,1,4,0,0,0,0,0,0,5,0,1,0},
```

```
{0,0,0,0,0,0,0,0,0,0,0,0},
{2,0,0,0,0,3,3,0,0,0,0,0,10},
{0,0,0,0,0,0,0,0,4,0,0,0},
}
```

The instance ID=6 is the example instance. A text format can be used to report a valid solution, as shown below:



The houses under protection should be marked, the number of families in those houses added up, and the police stations and their radius should be marked. Alternatively, the solutions can be presented graphically, via a spreadsheet, as used in this statement.

You must use blind searches to solve the e-folio, which means that you cannot order successors based on some kind of evaluation. You can use a fixed order that you consider best. You can't use any kind of heuristic information that evaluates the value of the state. It can exclude successors that it proves cannot lead to a solution.

You must deliver:

- Report;
- Source code for the implemented algorithms.

The report must contain a table with the results of running the algorithms/configurations tested against the instances provided. For each algorithm/instance it should show:

- Number of expansions;
- Number of generations;
- Result - impossible; solution; unresolved.
- Time spent (no more than 1 minute).

For each instance, in the result of each execution, one of three situations may occur: it obtains the proof that the instance is impossible; it solves the instance and in this case presents the solution; it does not solve it, exceeding the time limit. At the end, it must have the best information obtained considering all the runs, in other words, it presents the best it has obtained for each instance.

Template for the results table:

	Instance	1a	1b	…	10b
Algorithm 1 / settings 1	Expansions				
	Generations				
	Results				
	Time (msec)				
…					
	Best result				

In the appendix, you must present a solution obtained for each instance solved, the result of which was not impossible.

Correction criteria (4 points):

- **Analysis of the problem** (1 mark): Reference to important aspects of the problem in the report, revealing whether or not you implemented them, that you were aware of them, as well as the choices made in the implementation and their justification. Include in the analysis the branching, maximum depth and depth of solution of the search tree, elaborating on performance aspects of the algorithms you consider most appropriate.
- **Identification of algorithms** (1 mark): Clear identification of the algorithms you have implemented according to the nomenclature of the book and the CU, together with the configurations used, or if you use a different algorithm, you must describe it. It is possible to use another name for the same algorithms, as long as you indicate the corresponding one. The penalty for non-identification is 0.5 points.
- **Results** (2 points): This criterion is only assessed if there is a table of results, with each instance worth 0.1 marks. An instance is considered solved if a solution is obtained (with the solution correctly presented in an annex) or proof that the instance is impossible. You must indicate in the result what your executions allow you to conclude, and instances in which the correct conclusions have not been deduced from the executions carried out will be penalized by 0.05 points.

The work is individual, but if students wish, they can share results. Sharing results takes the problem away from a real situation, where you don't know how far you can go without references, but it can contribute to greater participation in the e-folio, and in no way affects the assessment. The results obtained by solving the example will be known after the grades have been posted.