Theory of Computer Games, NTU
Homework #1

Due date: 23:59 (UTC+8), November 10, 2016

Homework Description

- In this homework, you are asked to
  - Implement a solver of **Nonogram**
  - Compare the performance of different search algorithms

Nonogram

									1						1				
									1	2	2		2	2	1				
						3	2	1	6	1	1	7	1	1	6	1	2	3	
					5	5	9	11	2	3	3	1	3	3	2	11	9	5	5
					5	3	2	1	1	3	3	3	3	3	1	1	2	3	5
			5	5															
		3	5	3															
		2	9	2															
1	2	1	2	1															
		1	11	1															
		4	1	4															
		4	1	4															
				13															
			6	6															
				13															
	1	2	2	1															
		1	11	1															
		2	9	2															
		3	5	3															
			5	5															

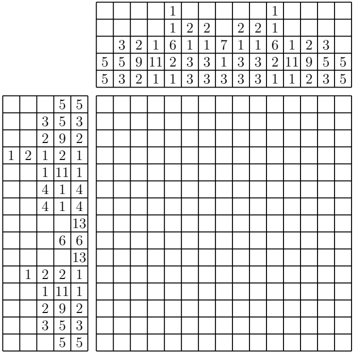
									1						1				
									1	2	2		2	2	1				
						3	2	1	6	1	1	7	1	1	6	1	2	3	
					5	5	9	11	2	3	3	1	3	3	2	11	9	5	5
					5	3	2	1	1	3	3	3	3	3	1	1	2	3	5
			5	5	■	■	■	■	■						■	■	■	■	■
		3	5	3	■	■	■			■	■	■	■	■			■	■	■
		2	9	2	■	■		■	■	■	■	■	■	■	■	■		■	■
1	2	1	2	1	■		■	■				■				■	■		■
		1	11	1	■		■	■	■	■	■	■	■	■	■	■	■		■
		4	1	4		■	■	■	■			■			■	■	■	■	
		4	1	4		■	■	■	■			■			■	■	■	■	
				13		■	■	■	■	■	■	■	■	■	■	■	■	■	
			6	6		■	■	■	■	■	■		■	■	■	■	■	■	
				13		■	■	■	■	■	■	■	■	■	■	■	■	■	
	1	2	2	1	■		■	■								■	■		■
		1	11	1	■		■	■	■	■	■	■	■	■	■	■	■		■
		2	9	2	■	■		■	■	■	■	■	■	■	■	■		■	■
		3	5	3	■	■	■			■	■	■	■	■			■	■	■
			5	5	■	■	■	■	■						■	■	■	■	■

- For each column and row, there is a hint
- For example, 5 5 means
  - There are 2 connected blocks strings
  - Each one has 5 connected blocks

Random Problem Generator

- Random problem generator from TCGA 2016
  - TCGA2016
  - http://aigames.nctu.edu.tw/ hsuehch/nonogram/tcga2016/boardgen.py
- Usage:
  - ./boardgen.py n num P1 P2 SEED
  - ./boardgen.py 25 1000 0.5 0.35 12345
    - n: size of board is n by n
    - num: number of test case
    - p1: max probability a cell is black
    - p2: min probability a cell is black
    - SEED: random seed

Input/Output



				1						1				
				1	2	2		2	2	1				
	3	2	1	6	1	1	7	1	1	6	1	2	3	
5	5	9	11	2	3	3	1	3	3	2	11	9	5	5
5	3	2	1	1	3	3	3	3	3	1	1	2	3	5

			5	5
		3	5	3
		2	9	2
1	2	1	2	1
		1	11	1
		4	1	4
		4	1	4
				13
			6	6
				13
	1	2	2	1
		1	11	1
		2	9	2
		3	5	3
			5	5

- Input:
  - $1 // Problem Number
  - 5 5 // Hint of first column, from up to down
  - ⋮
  - 5 5 // Hint of last column
  - 5 5 // Hint of first row, from left to right
  - ⋮
  - 5 5 // Hint of last row
- Output:
  - 1: block
  - 0: non-block

Standard Test Board

- Your program should at least pass the following test data
  - boardgen.py 5 10 0.5 0.3 12345
  - boardgen.py 10 10 0.5 0.3 12345
  - boardgen.py 15 10 0.5 0.3 12345
- Notification
  - The test data do not indicate the problem size
  - You can mention it from the argument list
  - e.g. “./solver 5 DFS < testboard”

Solution Package

- Submit page: http://w.csir.org/ tcg/2016/
- Code + documents to explain various heuristic used.
- Package structure:
  - Your ID [R05xxxxxx/B02xxxxxx/...]
    - code // A folder contains all your codes
    - report.pdf // Your report
- Compress your folder into a "zip" file
- Due to server limitation, the file size is restricted to 2M bytes

What Should Be Inclued in Your Reprot?

- About your code
  - How to compile and run your program.
  - What algorithm and heuristic you implement
- Experiment
  - The comparison bettwen different algorithms
- Discussion
  - The game complexity analysis
  - The factors affect the performance of each algorithm
  - The factors affect the difficulty of Nonogram
  - Other observation or discussion

References

- Nonogram's wikipedia page
  - https://zh.wikipedia.org/wiki/Nonogram
- TCGA 2016 Nonogram Tournament
  - http://http://aigames.nctu.edu.tw/ hsuehch/nonogram/tcga2016/
- An on-line nonogram playing site
  - http://www.puzzle-nonograms.com/