TCG Homework 1

徐銘威 R05944023

How to compile and run the program

1. Use g++ *.cpp to compile the programs.

2. Use ./*.out or *.exe [question_file] [answer_file] to run the programs.

What algorithm and heuristic to implement

My algorithm is basically based on the DFS. Use the row hint to generate the possible row combination set and check them if the filled region satisfied the column hint.

1. DFS_Single.cpp : The DFS_single algorithm would only use heuristic at the beginning, making a guess board for the generated combination(both row and column) and eliminating the impossible combinations.

2. DFS_pre.cpp : The DFS_pre algorithm would do a pre-elimination after filling a row, which would eliminate the impossible combination. This pre_preprocessing is just like making a guess board for each layer and would terminate the impossible branch during the search.

The comparison between different algorithms

	DFS_Single	DFS_pre.cpp
n=5	0.005397	0.003748
n=10	1.22156	0.018167
n=15	>10 min	2.34657

The game complexity analysis

We can see Nonograms as finding either a box is 0 or 1 for the whole board. The rows and columns form 2*n n-SAT problem which is NP-complete (proof is omitted). The game complexity is 2^{n^2}.

However, from the clues we can find a series of combination for each row and column, which lower the game complexity to around b^n, where b is the maximum number of combination for a column or row.

The factors affect the performance of each algorithm

The factors the mainly affect the performance is the number of branches and depth. Since the depth is fixed to n, the algorithm would take a longer time if the branch is a lot. The depth is related to problem complexity.

Another factor that affect the performance of DFS with pre-processing is how much combination we can eliminate after doing pre-process. Those with high elimination rate of combinations give a better performance.

The last factor is how we save the data of the combinations, because we need to do a lot of processing in each DFS step. The pre_elimination would be influenced more. So use the simple structure would get the better performance.

The factors affect the difficulty of Nonogram

The main factor that affect the difficulty of Nonogram is the problem complexity which is the size of n.

Another huge factor is about the combinations. We can observe that some clues give a very limited number of possible combinations but some give a lot, especially those that is medium sparse. Noted in here we consider the space after black cells comes with a fixed white cell after it, the number of combinations is to fill white spaces in between and at the begin or end of these blocks, which is maximum in between 1/3 to 1/2 depends on size of each block.