Assignment 1 Solution and Marking Scheme

1. Informed Search (40 points)

1(a) (6 points: 2 for the correct answer and 4 for the proof)

Consider H_1 = *Misplaced Tile Heuristic* and H_2 = Manhattan Distance *Heuristic*. Acceptable "proofs" should have that H_2 is better than H_1 because H_2 is closer to the correct cost, *i.e.*, $H^*(n) \geq H_2(n) \geq H_1(n)$, where $H^*(n)$ is the real cost. This implies that the time taken to do search should be lower. In other words, H_2 provides a tighter lower bound.

1(b) (7 points for each heuristic: 3 points for correct answer and 4 points for poof)

For a heuristic to be consistent, it must comply with the following:

$$h(n) \leq c(n,n') + h(n') \ \forall n,n'$$

<u>Proposition:</u> H_1 is consistent

<u>Proof:</u> 3 cases need to be addressed:

1 - After a move, the SAME NUMBER of tiles is out of position:

In this case, $c(n,n') = 1$ and $H(n) = H(n')$. So, $h(n) \leq c(n,n') + h(n')$.

2 - After a move, one LESS tile is out of position:

In this case, $c(n,n') = 1$ and $H(n) = H(n') -1$. So, $h(n) \leq c(n,n') + h(n')$.

3 - After a move, one MORE tile is out of position:

In this case, $c(n,n') = 1$ and $H(n) = H(n') +1$. So, $h(n) \leq c(n,n') + h(n')$.

These 3 cases show that no matter the move, we will have $h(n) \leq c(n,n') + h(n')$, which means that after each move the *f-cost* increases or stays the same. Therefore, the *f-cost* increases monotonically.

<u>Proposition:</u> H_2 is consistent

The proof for this proposition is almost identical to the proof given above, with the exception that the first case is not possible anymore. When the blank moves, H_2 's value is either *-1* or *+1* from *n* to *n'*. Note: An answer like this one was given full marks.

2(a) (3 points for each)

Time complexity: $O(b^d)$;
Space complexity: $O(bd)$.

2(b) (2 points for the correct answer, 5 points for the proof. Note: 2 points will be deducted if no explanation about the fact that it will not go down the tree indefinitely.)

*IDA** is complete because the algorithm uses a *DFS* scheme to visit exhaustively the tree of solutions while increasing continually the fringe depth limit after each iteration, until the goal is reached. This limit stops the search from going down the tree indefinitely.

Note 1: The fact that the fringe ceiling is derived from an *f-cost* value doesn't change this assumption as long as the branching factor is finite, and that the costs are positive(greater than some constant).

Note 2: This is only true if we assume the computer has enough resources available (memory).

2(c) (2 points for correct answer, 5 points for proof, if proof is not detailed or quite to the point, 3 points will be deducted.)

*IDA** is optimal because it uses a *DFS* scheme similar to *IDS*, and *IDS* is optimal. Every time *IDA** chooses a new *cutoff* value, it chooses a value that is larger than its previous *cutoff*, but that is the smallest value among all the *f-cost* values encountered so far. Hence, the optimality relies upon the consistency of the heuristic in use, which guarantees that the *f-costs* are monotonically increasing by some 'optimistic' amount.

2. Constraint Satisfaction (60 points)

1. (15 points: 5 points for each question)

Example of a possible solution:

Variables: $V_1 ,...,V_{81}$
Domain: $\{0,...,9\}$
Constraints:

- Each row contains integers from 1 to 9 without duplications;
- Each column contains integers from 1 to 9 without duplications;
- Each box contains integers from 1 to 9 without duplications.

2. (45 points: 15 points will be given for the explanation. 10 points in total for each category. If the number of steps is incorrect (much higher than expected), 2 points will be deducted for each category.

Example of an acceptable solution for time and number of steps (nodes):

	B		B + FC		B + FC + H	
Easy	31 ms	2550	63 ms	1053	32 ms	52
Medium	16 ms	484	31 ms	292	31 ms	52
Difficult	94 ms	13173	88 ms	1912	109 ms	155
Evil	89 ms	12974	113 ms	2579	53 ms	65

Note: Clearly, these values depend on the implementation.