CSE-537 INTRODUCTION TO ARTIFICIAL INTELLIGENCE

ASSIGNMENT-2 REPORT

NAME: SNIGDHA KAMAL STUDENT ID: 110937472

SECTION-I

1. DFSB

S. No	Test case	Time Taken	Number of search calls	Number of arc pruning calls
1.	Backtrack_easy	0.004 seconds	8	0
2.	Backtrack_1	60 seconds (No answer)	3151485	0
3.	Backtrack_2	60 seconds (No answer	2732350	0

2. DFSB+

S. No	Test case	Time Taken	Number of search calls	Number of arc pruning calls
1.	Backtrack_easy	0.006 seconds	8	13
2.	Backtrack_1	0.393 seconds	151	539
3.	Backtrack_2	0.99 seconds	208	941

3. Min-Conflicts

S. No.	Test case	Time Taken	Number of search steps
1.	Minconflict_easy	0.073 seconds	221
2.	Minconflict_1	0.018 seconds	101
3.	Minconflict_2	0.013 seconds	77

SECTION-II

OBSERVATIONS

1. **Depth First search with Backtracking:**
   This is an exhaustive search method which expands all the nodes in its path without any pruning. Thus, this algorithm is effective only when the search space is small (number of vertices in the graph is low). DFSB only gave an output for backtracking_easy.txt test case as the number of vertices was low. It failed to yield a solution for the other test cases in the stipulated time

2. **Depth First Search with Backtracking++:**
   This is a smart algorithm which effectively performs pruning and reduces the search space using three heuristics:

   - **Minimum Remaining Values(MRV)**
     This heuristic chooses the variable which has the least number of legal values left. This way, it chooses the variable which will fail-first if not chosen next. This also reduces the branching factor of the search
   - **Least Constraining Value(LCV)**
     Once the variable to be assigned next is chosen, this heuristic chooses the value for that variable which least constrains its neighbours. This works because a value that doesn't affect its neighbours is least likely to cause failure and a value that restricts the domains of its neighbours the most is highly likely to cause failure.
   - **AC-3 Algorithm (Arc Consistency Algorithm)**

     AC-3 Algorithm is an extension of the forward checking algorithm. It is used to perform an inference of how the domains of neighbours and the vertices connected to the neighbours will be affected if a value is assigned to a vertex. If the domain of any vertex in the graph becomes empty, then inference is false and the vertex is assigned a new value and the AC-3 algorithm is applied again

     In my code, DFSB++ was easily able to solve all three test cases in less than one second while the plain DFSB was unable to solve backtrack_1 and backtrack_2 even in 60 seconds. The arc pruning calls is the number of times each edge is checked for consistency

3. **Min Conflicts Algorithm**
   Min-conflicts is a local search algorithm based on randomization. It randomly chooses any conflicting variable, i.e., the variable involved in any unsatisfied constraint, and then picks a value which minimizes the number of violated constraints.
   The problem I faced was that my code kept getting stuck in a local minimum. I used Random Walk strategy to break out of this by either assigning the variable the colour that minimizes the conflicts with a probability p or a random value with probability 1-p.
   Using this, my code could come out of the local minima.
   The code yields result for all three test cases in under 1 second

REFERENCES:

- Constraint Satisfaction class slides
- Introduction to Artificial Intelligence: A Modern Approach
- Algorithms discussed with Naveen Rai