Writeup Part

## Question 1)

1. Since we just need one variable to express whether a person is going on the car or not and we have six people, we just need 6 variables. Since there are two choices for each person, there are assignments in total.
2. a)

b)

c) , which implies

d) which implies

which implies

So, overall, we have the CNF of:

1. Yes it is satisfiable. One such solution is are true. are false.

## Question 2)

1. We implemented the neighborhood functionality outside of Tabu.m. Please refer to the code attached/sent in the email.
2. Since there are twenty variables and each variable can take on two possible choices, there are variable assignments for this problem.
3. Since k is at most twenty, a relatively small number, we can just run the experiment with all possible values of k and determine the best possible k. In cases where k can take on more values, we could use greedy search or another algorithm to find k.



We ran the experiment of looping over possible k, and output the average cost per same 100 initial conditions for each k, and plotted the above graph. From the graph, we can see the best solution is k=8.

1. We picked 100 as the number of iterations because this allows us to achieve the optimal solution over 95% of the time. We also picked k=8 as stated in part c).
   1. After running this many times, we found that about 97-99 of the 100 trials reached the optimal solution with iteration number = 100. We did not see more than one satisfiable solution.
   2. The average number of iterations needed to find the satisfying solution is about 24.9. Since we evaluate the cost of 20 neighbors in each iteration, the number of cost function evaluations required on average to find the satisfying solutions is around .
   3. For trials that do not find satisfying solutions, they must have gone through every iteration. Since we set the maximum iteration as 100, they will evoke cost evaluations. The average value of our best objective function over 100 trials is 0.000220. (Notice that we define the cost of an assignment as the ratio of number of unsatisfied clauses over the number of total clauses. In that setup, the cost value is between 1 and 0, and the lower the cost is, the better the solution is. Global optimum is reflected as the assignment cost being 0.)

## Question 3)

1. The permutation of the whole domain of 10 elements is: = 10! = 3628800
2. The size of neighborhood is 9, since once we picked the position, there are 9 other positions that this one can be swapped with.
3. There are 9 neighbors. They are:
   1. (8 9 7 6 5 4 3 2 1 0)
   2. (7 8 9 6 5 4 3 2 1 0)
   3. (6 8 7 9 5 4 3 2 1 0)
   4. (5 8 7 6 9 4 3 2 1 0)
   5. (4 8 7 6 5 9 3 2 1 0)
   6. (3 8 7 6 5 4 9 2 1 0)
   7. (2 8 7 6 5 4 3 9 1 0)
   8. (1 8 7 6 5 4 3 2 9 0)
   9. (0 8 7 6 5 4 3 2 1 9)
4. After you finish making 10 swaps (where 10 is the number of elements), you wrap around to the beginning, so essentially during the iteration, you are swapping at position . Therefore, during the iteration, you will swap at position 1.
5. The cycle length is 9. In this example, we consider the sub-neighborhood as all swaps involving position. Then, after 9 iterations, we would have already considered all swaps involving all positions. Since for the last position, each of its possible swaps is covered in the first 9 iterations respectively.
6. The advantage of using cycling scheme is that it provides a reduced neighborhood, and therefore reduces runtime per each iteration without sacrificing too much efficiency(each neighborhood structure would be considered eventually).
7. If we could evaluate all possible permutations in iteration 1, then we would only need only one iteration in total. Because during first iteration, we would have already tried all possible permutations and have calculated their cost, simply picking the one with optimal cost yields the global optimal solution.