Homework Assignment #18 Key

Question 1: Give the reduced cost matrix and the lower bound for the following distance matrix.

∞	7	3	12
3	∞	6	14
5	8	∞	6
9	3	5	∞

We first reduce the rows

				Cost
∞	4	0	9	3
0	∞	3	11	3
0	3	∞	1	5
6	0	2	∞	3
				14 = Total cost for row reduction

We then reduce columns with only the last column requiring reduction with cost 1 for a total cost bound of 15.

∞	4	0	8
0	∞	3	10
0	3	∞	0
6	0	2	∞

Question 2: 9.3 from the book.

a) There are multiple variations. Two possible approaches are: 1) let the initial state include no sets with an initial lower bound. A subproblem is a child created by adding one of the remaining sets to those of the parent. Another approach is to divide a state into two children, one which includes a set and the other which excludes the set (like the include/exclude version of TSP).

b) Any state on the Queue with a lower bound less than BSSF. Typically choose the state with the lowest lower bound.

c) For the first variation above, you create a child state for each remaining set which can be added to the parent state. For the second variation, you choose one remaining set and create two children, one including and the other excluding the set. For this case you should try to be smart about which sets you try first, such as try the biggest sets or the sets with the most uncovered elements for the include side.

d) The lower bound is an optimistic estimate on the number of sets required. There are multiple possibilities. You can't just use 0. Here is a greedy one. The parent state already includes a certain number of sets c which cover a subset of the vertices. Call the set of remaining vertices to be covered V. Choose the remaining set which covers the largest number of elements of V. The number of elements covered is v1. Then choose the remaining set that covers the next largest number of elements of V and that number is v2 (note that we did not subtract from V the v1 elements covered by the first chosen set). Repeat until $v1 + v2 +\_ \ldots + vn \geq |V|$. The lower bound is c + n. The optimal number of sets cannot be less than this.