Traveling Salesman Problem (G)

```
//assume that the n cities are v1, v2, .., vn
n = |V|
minCost = ∞
minTour = ∅
//without loss of generality, assume that all tours start from v1
O((n-1)!) for each permutation of vertices v2, v3, v4, .., vn
              //compute the cost of the tour
              crtCost = 0
          O(n) for i = 2 to n-1
                  crtCost = crtCost + w(vi, vi+1)
              if crtCost < minCost
                  minCost = crtCost
                  minTour = (v1, v2, v3, ..., vn)
    return minTour
```

Total RT $= O((n-1)! \cdot n) = \underline{O(n!)}$

$$\boxed{RT = O(n!)}$$

KnapsackProblem (v, w, W)

```
       if v.length ≠ w.length
           return error
       n = v.length   // n is the number of items
       maxValue = 0
       maxValueSubset = Ø
O(2^n) for each subset S of items
           totalWeight = 0
           totalValue = 0
    O(n)   for i = 1 to n
               if item i ∈ S
                   totalWeight = totalWeight + w[i]
                   totalValue = totalValue + v[i]
           if totalWeight ≤ W and totalValue > maxValue
               maxValue = totalValue
               maxValueSubset = S
       return maxValueSubset
```

$\boxed{RT = O(n \cdot 2^n)}$

cost matrix

Assignment Problem (c)

```
       // c has size n x n
       n = c.size
       minCost = ∞
       minCostAssign = Ø
O(n!)  for each permutation j1, j2, j3, ..., jn
           crtCost = 0
    O(n)   for i = 1 to n
               crtCost = crtCost + c[i, ji]
           if crtCost < minCost
               minCost = crtCost
               minCostAssign = j1, j2, ..., jn
       return minCostAssign
```

$\boxed{RT = O(n \cdot n!)}$