Department of Electrical and Computer Engineering
EECE 4542: Computer Engineering Algorithms

Solutions to Homework #3

Solve the following problems.

1. In Maximum Weight Matching problem, we are given a weighted graph and we are looking for a matching where the selected edges have maximum total weight. For the graph in Figure 1, the weight of each edge ei = i. Find a solution to this problem.

   The set $\{e2, e5, e7\}$ *is optimal.*

2. Describe a greedy approach to solving Maximum Weight Matching, and apply it to the graph in Figure 1. Describe an instance of the problem that your greedy algorithm does not solve correctly, and explain why the algorithm fails.

   A greedy approach is to always add the edge with the largest weight that does not conflict with previously added edges. The greedy approach would select edge e7, then e4, then e2, for a total weight of 13 which is optimal.

   An example where the greedy algorithm fails is:

   ```
   (1,2) weight = 1
   (2,3) weight = 3
   (3,4) weight = 5
   (4,1) weight = 4
   ```

3. Explain why a greedy approach to solving the Knapsack problem based on the ratio of the value to the cost of each item makes sense. Describe an instance of the problem that this approach does not solve correctly, and explain why the algorithm fails.

   The ratio of the value to the cost of an item is the value density of the item – i.e. it is value per unit cost. It makes sense to select objects with a large value density, because they give the maximum value for the minimum cost.

   This algorithm fails to correctly solve the following instance:

   ```
   Object 1: value = 3, cost = 4
   Object 2: Value = 2, cost = 3
   Object 3: Value = 2, cost = 3
   Cost bound: 6
   ```

   The algorithm selects Object 1, but cannot select another object giving a total value of 3. *However an optimal subset includes Objects* 2 *and* 3, *giving a total value of* 4.

4. Describe an instance of the Activity Assignment problem that is not correctly solved by the following algorithm: Order the activities by increasing start times, consider the activities one by one in order, add an activity if it does not conflict with any previously added activity.

   The algorithm fails to correctly solve the following instance:

   ```
   Activity 1: start = 1, end = 4
   Activity 2: start = 2, end = 3
   Activity 3: start = 3, end = 4
   ```

The algorithm would select only Activity 1, but an optimal subset includes Activities 2 and 3.

5. For the Maximum Matching problem applied to the bipartite graphs in Figures 2 and 3:

   (a) given the matching A in each case, give augmenting paths containing three and five edges, if they exist,

   For Figure 2, the augmenting paths are $(e6, e1, e3)$ *and* $(e7, e5, e4, e1, e3)$. *Figure 3 does not have an augmenting path containing three edges, but* $(e10, e1, e3, e9, e8)$ *is an augmenting path containing five edges.*

   (b) given the matching A in each case, show the vertices visited in the breadth-first search for an augmenting path,

   The edges visited are shown in Figures 2 and 3.

   (c) give the corresponding improved matching.

   For Figure 2, augmenting path $(e6, e1, e3)$ *gives the new matching* $\{e6, e3, e5, e9\}$. *For Figure 3, augmenting path* $(e8, e9, e3, e1, e10)$ *gives the new matching* $\{e5, e6, e8, e3, e10\}$.