

# Homework 3

Zach Neveu

May 28, 2019

## 1. Maximum Weighted Matching

- {e7, e5, e2}

## 2. Greedy Weighted Matching

- Start with the highest weighted edge. Progressing through the edges in decreasing order of weight, add each edge if it does not conflict with the current matching.
- On Figure 1, if e1 was swapped with e5, this algorithm would fail. By choosing the highest weighted edge, this algorithm fails to account for the fact that a combination of lower weighted edges may add to a higher total than the heaviest edge.

## 3. Greedy Knapsack vs. Matching

- In knapsack, selecting one item does not disqualify any specific items from membership. In matching, by contrast, including a certain edge disqualifies neighboring edges. For this reason, a greedy algorithm makes more sense to solve knapsack than to solve matching.
- One instance of knapsack where a greedy algorithm would fail is if the costs of each item are very large in comparison to the cost limit, and the value to cost ratios of all items are similar. In this case, selecting a high value-to-cost item that does not come close to the cost limit could be worse than selecting many lower-value items that get closer to the cost limit.

## 4. Activity Scheduling

- Imagine an activity that lasts all day. It starts earlier than any activity and ends later than any activity. The algorithm described will select this activity because it starts first. Selecting this activity does not allow for any other activities selected, which obviously yields a sub-optimal result.

## 5. Bipartite Matching

- Fig 2: {e6, e1, e3}, {e7, e5, e4, e1, e3}. Fig 3: none, {e8,e9,e3,e1,e10}
- Fig 2: {a1->b4, b2->a3}, Fig 3: {a1->b4, b2->a5, b4->a4, a4->b1}
- Fig 2: {e3,e4,e7,e8}, Fig 3: {e8,e3,e10,e5,e6,}

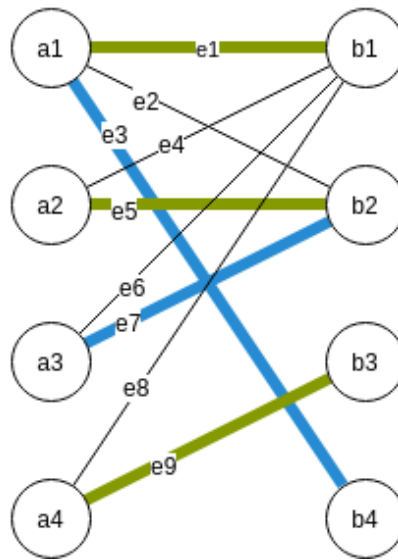


Figure 1: First Bipartite Matching Example

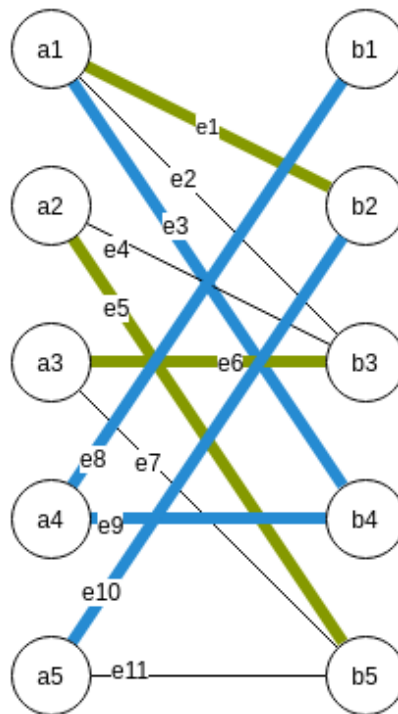


Figure 2: Second Bipartite Matching Example