EECS 336 Fall 2015
Homework Problem 3.1

Theorem: Greedy by Value algorithm produces optimal solution if and only if the output sets are matroid. Re-state Greedy by Value for a general matroid $\mathcal{M} = (\mathcal{E}, \mathcal{I})$, with E as the ground set and w_i as weight for each element $i \in \mathcal{I}$.

Algorithm 1 Greedy by Value algorithm for $\mathcal{M}$

```
procedure GreedyByValue(M, w_i)
    Initialize S = ∅
    Sort w_i by descending values: w_1 ≥ w_2 ≥ ... ≥ w_n
    for i = 1 to n do
        if S + i ∈ I then
            S ← S + i
        end if
    end for
    return S
end procedure
```

Case 1: Suppose $\mathcal{M}$ violates downward closure, and there exists some $S \subset T$, $T \in \mathcal{I}$ but $S \notin \mathcal{I}$. We assign the weights such that $w_i = w_a$ for every $i \in S$, $w_j = w_b$ for $j \in T \backslash S$, and $w_k = 0$ for $k \in E \backslash T$, with $w_a > w_b > 0$.

Suppose the algorithm selects a subset $S_1 \subset S$ as part of the solution. Since $S \notin \mathcal{I}$, S_1 cannot be same as S. Let S_2 be the entire set of elements that the algorithm selects. Then the total weights of elements selected by the algorithm is:

$$
\begin{aligned}
&W(S_2) = \sum_i w_i(i \in S_1) + \sum_j w_j(j \in S_2 \backslash S) \\
&< w_a\,|S| + w_b\,|T \backslash S| \\
&= W(T)
\end{aligned}
$$

This indicates that the optimal solution from the algorithm is worse than taking the sum of weights from elements in T, which contains S and has to obey downward closure. Thus, when downward closure is not satisfied, algorithm 1 always provides suboptimal solution that is smaller than $W(T)$.

Case 2: Suppose the set system is not a matroid because the augmentation property does not hold. In particular, let $S, T \in \mathcal{I}$ be two independent sets and $|S| < |T|$. For all $i \in T \backslash S$, $S + i \notin \mathcal{I}$. To formulate a counter example, we start with the following weights: $w_i = w_1$ for $i \in S$;$w_i = w_2$ for $i \in T \backslash S$; $w_i = w_3 = 0$ otherwise. $(w_1 > w_2 > w_3)$

Since S is in $\mathcal{I}$, by greedy S will be all selected. Since $\forall i \in T \backslash S$, $S + i \notin \mathcal{I}$, none of the elements in $T \backslash S$ would be chosen. (Note $w_3 = 0$, the selection in $E \backslash (T \cup S)$ won't influence the total weight.)

$W(greedy) = W(S) = w_1|S|$

$W(T) = w_2|T \backslash S| + w_1|T \cap S|$

The counterexample aims to show that in some certain settings, $W(greedy)$ is not maximum. Suppose we let $W(T) > W(greedy)$, then below relations must hold:

$w_2|T \backslash S| + w_1|T \cap S| > w_1|S|$

$w_2|T| - w_2|T \cap S| > w_1|S| - w_1|T \cap S|$

Therefore, weights satisfies $w_1 > w_2 > w_1|S \backslash T|/|T \backslash S|$ could be the counterexamples.

Now we let $w_2 = \frac{|S|}{|T|}w$, $w_1 = w$, where w is a positive constant.

$W(S) = |S|w$

$W(T) = \frac{|S|}{|T|}w \times |T - T \cap S| + w|T \cap S|$

$= \frac{|S|}{|T|}w \times |T| + (w|T \cap S| - \frac{|S|}{|T|}w \times |T \cap S|)$

Since $|S| < |T|$, the terms in the parentheses is greater than zero. So,

$W(T) > \frac{|S|}{|T|}w \times |T| = |S|w = W(S) = W(greedy)$

So for all cases within this setting, greedy is not optimal.

Case 3: Suppose both downward closure and augmentation theorem is violated in the set system, then either of above cases can be used to show that greedy algorithm does not provide optimal solution.