19.5
Maximum Weight Subset of Elements: Cardinality and Beyond

Picking k elements to maximize total weight

1. Given n items each with non-negative weights/profits and integer $1 \leq k \leq n$.
2. Goal: pick k elements to maximize total weight of items picked.

	e_1	e_2	e_3	e_4	e_5	e_6
weight	3	2	1	4	3	2

$k = 2$:
$k = 3$:
$k = 4$:

Greedy Template

```
N is the set of all elements X ← ∅
(* X will store all the elements that will be picked *)
while |X| < k and N is not empty do
    choose e_j ∈ N of maximum weight
    add e_j to X
    remove e_j from N
return the set X
```

Remark: One can rephrase algorithm simply as sorting elements in decreasing weight order and picking the top k elements but the above template generalizes to other settings a bit more easily.

Theorem 19.1.

Greedy is optimal for picking k elements of maximum weight.

Greedy Template

```
N is the set of all elements X ← ∅
(* X will store all the elements that will be picked *)
while |X| < k and N is not empty do
    choose e_j ∈ N of maximum weight
    add e_j to X
    remove e_j from N
return the set X
```

Remark: One can rephrase algorithm simply as sorting elements in decreasing weight order and picking the top k elements but the above template generalizes to other settings a bit more easily.

Theorem 19.1.

Greedy is optimal for picking k elements of maximum weight.

A more interesting problem

1. Given n items $N = \{e_1, e_2, \ldots, e_n\}$. Each item e_i has a non-negative weight w_i.
2. Items partitioned into h sets $N_1, N_2, \ldots, N_h$. Think of each item having one of h colors.
3. Given integers $k_1, k_2, \ldots, k_h$ and another integer k
4. Goal: pick k elements such that no more than k_i from N_i to maximize total weight of items picked.

	e_1	e_2	e_3	e_4	e_5	e_6	e_7
weight	9	5	4	7	5	2	1

$N_1 = \{e_1, e_2, e_3\}$, $N_2 = \{e_4, e_5\}$, $N_3 = \{e_6, e_7\}$
$k = 4$, $k_1 = 2$, $k_2 = 1$, $k_3 = 2$

Greedy Template

```
N is the set of all elements X ← ∅
(* X will store all the elements that will be picked *)
while N is not empty do
    N' = {e_i ∈ N | X ∪ {e_i} is feasible}
    if N' = ∅ then break
    choose e_j ∈ N' of maximum weight
    add e_j to X
    remove e_j from N
return the set X
```

Theorem 19.2.

Greedy is optimal for the problem on previous slide.

Proof: exercise after class.

Special case of general phenomenon of Greedy working for maximum weight independent set in a matroid. Beyond scope of course.

Greedy Template

```
N is the set of all elements X ← ∅
(* X will store all the elements that will be picked *)
while N is not empty do
    N' = {e_i ∈ N | X ∪ {e_i} is feasible}
    if N' = ∅ then break
    choose e_j ∈ N' of maximum weight
    add e_j to X
    remove e_j from N
return the set X
```

Theorem 19.2.

Greedy is optimal for the problem on previous slide.

Proof: exercise after class.

Special case of general phenomenon of Greedy working for maximum weight independent set in a matroid. Beyond scope of course.

THE END

...

(for now)