Greedy
CSCI 432
Greedy Algorithms:

- Characterize structure of optimal solution.
- Recursively define value of optimal solution.
- Compute value of optimal solution.
- Construct optimal solution from computed information.

Dynamic Programming:

- Characterize structure of optimal solution.
- Recursively define value of optimal solution.
- Compute value of optimal solution.
- Construct optimal solution from computed information.
Greedy Algorithms:

Dynamic Programming:
- Characterize structure of optimal solution.
- Recursively define value of optimal solution.
- Compute value of optimal solution.
- Construct optimal solution from computed information.

Greedy:
- Make the choice that best helps some objective.
- Do not look ahead, plan, or revisit past decisions.
- Hope that optimal local choices lead to optimal global solutions.
Greedy Algorithms:

Dynamic Programming:
• Characterize structure of optimal solution.
• Recursively define value of optimal solution.
• Compute value of optimal solution.
• Construct optimal solution from computed information.

Greedy:
• Make the choice that best helps some objective.
• Do not look ahead, plan, or revisit past decisions.
• Hope that optimal local choices lead to optimal global solutions.

Greedy algorithm for:
• Robbing a jewelry store?
• Eating at a fancy buffet?
Proof of Optimality?

\[
E(i, j) = \min \left\{ \begin{array}{l}
E(i - 1, j) + 1 \\
E(i, j - 1) + 1 \\
E(i - 1, j - 1) + \text{diff}(i, j)
\end{array} \right.
\]

\[
\text{diff}(i, j) = \begin{cases} 
0, & x[i] = y[j] \\
1, & x[i] \neq y[j]
\end{cases}
\]
Proof of Optimality?

\( O_n = \text{optimal profit from partitioning rod of length } n. \)
\( p_i = \text{profit for rod of length } i. \)

\[
O_n = \max_{1 \leq i \leq n} (p_i + O_{n-i})
\]

\( C(p) \) – minimum number of coins to make \( p \) cents.
\( x \) – value (e.g. $0.25) of a coin used in the optimal solution.

\[
C(p) = \begin{cases} 
\min_{i: d_i \leq p} C(p - d_i) + 1, & p > 0 \\
0, & p = 0
\end{cases}
\]
Minimum Spanning Tree

Tree – connected graph with no loops.
Minimum Spanning Tree

Tree – connected graph with no loops.

Spanning tree – tree that includes all vertices in a graph.
Minimum Spanning Tree

Tree – connected graph with no loops.

Spanning tree – tree that includes all vertices in a graph.

Minimum spanning tree – spanning tree whose sum of edge costs is the minimum possible value.
Kruskal’s MST Algorithm

Greedy decision: Add the edge with smallest weight, that does not create a cycle.
Kruskal’s MST Algorithm

**Greedy decision:** Add the edge with smallest weight, that does not create a cycle.

![Graph](image-url)
Kruskal’s MST Algorithm

Greedy decision: Add the edge with smallest weight, that does not create a cycle.
Kruskal’s MST Algorithm

**Greedy decision:** Add the edge with smallest weight, that does not create a cycle.
Kruskal’s MST Algorithm

Greedy decision: Add the edge with smallest weight, that does not create a cycle.
Kruskal’s MST Algorithm

**Greedy decision:** Add the edge with smallest weight, that does not create a cycle.
Kruskal’s MST Algorithm

Greedy decision: Add the edge with smallest weight, that does not create a cycle.
Kruskal’s MST Algorithm

**Greedy decision:** Add the edge with smallest weight, that does not create a cycle.

**Proof of validity:** ?
Kruskal’s MST Algorithm

Greedy decision: Add the edge with smallest weight, that does not create a cycle.

Proof of validity: Let $G = (V, E)$, and $T \subseteq E$ be the set of edges resulting from Kruskal’s algorithm.
Kruskal’s MST Algorithm

Greedy decision: Add the edge with smallest weight, that does not create a cycle.

Proof of validity: Let $G = (V, E)$, and $T \subseteq E$ be the set of edges resulting from Kruskal’s algorithm.

What do we need to show?
Kruskal’s MST Algorithm

**Greedy decision:** Add the edge with smallest weight, that does not create a cycle.

**Proof of validity:** Let $G = (V, E)$, and $T \subseteq E$ be the set of edges resulting from Kruskal’s algorithm.

$T$ is a tree because it is connected (otherwise we could have added more edges without creating cycles) and there are no cycles.
Kruskal’s MST Algorithm

Greedy decision: Add the edge with smallest weight, that does not create a cycle.

Proof of validity: Let $G = (V, E)$, and $T \subseteq E$ be the set of edges resulting from Kruskal’s algorithm.

$T$ is a tree because it is connected (otherwise we could have added more edges without creating cycles) and there are no cycles.
Kruskal’s MST Algorithm

**Greedy decision:** Add the edge with smallest weight, that does not create a cycle.

**Proof of validity:** Let $G = (V, E)$, and $T \subseteq E$ be the set of edges resulting from Kruskal’s algorithm.

$T$ is a tree because it is connected (otherwise we could have added more edges without creating cycles) and there are no cycles.

$T$ spans $G$ because if it did not, we could have added more edges to connected unreached nodes without creating cycles.
Kruskal’s MST Algorithm

Greedy decision: Add the edge with smallest weight, that does not create a cycle.

Proof of validity: Let $G = (V, E)$, and $T \subseteq E$ be the set of edges resulting from Kruskal’s algorithm.

$T$ is a tree because it is connected (otherwise we could have added more edges without creating cycles) and there are no cycles.

$T$ spans $G$ because if it did not, we could have added more edges to connected unreached nodes without creating cycles.
Kruskal’s MST Algorithm

Greedy decision: Add the edge with smallest weight, that does not create a cycle.

Proof of validity: Let $G = (V, E)$, and $T \subseteq E$ be the set of edges resulting from Kruskal’s algorithm.

$T$ is a tree because it is connected (otherwise we could have added more edges without creating cycles) and there are no cycles.

$T$ spans $G$ because if it did not, we could have added more edges to connected unreached nodes without creating cycles.

$\therefore T$ is a spanning tree of $G$
Kruskal’s MST Algorithm

**Greedy decision:** Add the edge with smallest weight, that does not create a cycle.

**Proof of optimality:** ?
Kruskal’s MST Algorithm

Greedy decision: Add the edge with smallest weight, that does not create a cycle.

Proof of optimality: $T$ is an MST, because???
MST Cut Property

Lemma: Suppose that $S$ is a subset of nodes from $G = (V, E)$. Then, the cheapest edge $e$ between $S$ and $V \setminus S$ is part of every MST.

Proof:
MST Cut Property

Lemma: Suppose that $S$ is a subset of nodes from $G = (V, E)$. Then, the cheapest edge $e$ between $S$ and $V \setminus S$ is part of every MST.

Proof:
MST Cut Property

Lemma: Suppose that $S$ is a subset of nodes from $G = (V, E)$. Then, the cheapest edge $e$ between $S$ and $V \setminus S$ is part of every MST.

Proof:
Lemma: Suppose that $S$ is a subset of nodes from $G = (V, E)$. Then, the cheapest edge $e$ between $S$ and $V \setminus S$ is part of every MST.

Proof: Any MST of $G$ must include some edge between $S$ and $V \setminus S$ (otherwise it would not be a tree).
**MST Cut Property**

**Lemma:** Suppose that $S$ is a subset of nodes from $G = (V, E)$. Then, the cheapest edge $e$ between $S$ and $V \setminus S$ is part of every MST.

**Proof:** Any MST of $G$ must include some edge between $S$ and $V \setminus S$ (otherwise it would not be a tree).

Let $e$ be the cheapest edge between $S$ and $V \setminus S$.  

![Diagram of a graph with sets $S$ and $V \setminus S$ and edges $e'$ and $e$]
MST Cut Property

Lemma: Suppose that $S$ is a subset of nodes from $G = (V, E)$. Then, the cheapest edge $e$ between $S$ and $V \setminus S$ is part of every MST.

Proof: Any MST of $G$ must include some edge between $S$ and $V \setminus S$ (otherwise it would not be a tree).

Let $e$ be the cheapest edge between $S$ and $V \setminus S$.

Suppose $T$ is a ST that does not include $e$. 

\[ e' \]
MST Cut Property

Lemma: Suppose that $S$ is a subset of nodes from $G = (V, E)$. Then, the cheapest edge $e$ between $S$ and $V \setminus S$ is part of every MST.

Proof: Any MST of $G$ must include some edge between $S$ and $V \setminus S$ (otherwise it would not be a tree).

Let $e$ be the cheapest edge between $S$ and $V \setminus S$.

Suppose $T$ is a ST that does not include $e$. Then:

1. $T \cup \{e\}$ must have a cycle. Because?

2. That cycle must have another edge $e'$ between $S$ and $V \setminus S$. Because?
MST Cut Property

Lemma: Suppose that \( S \) is a subset of nodes from \( G = (V, E) \). Then, the cheapest edge \( e \) between \( S \) and \( V \setminus S \) is part of every MST.

Proof: Any MST of \( G \) must include some edge between \( S \) and \( V \setminus S \) (otherwise it would not be a tree).

Let \( e \) be the cheapest edge between \( S \) and \( V \setminus S \).

Suppose \( T \) is a ST that does not include \( e \). Then:

1. \( T \cup \{e\} \) must have a cycle. (Since spanning tree \( T \) already has a path between \( u \) and \( v \), so adding \( e \) will create a cycle)

2. That cycle must have another edge \( e' \) between \( S \) and \( V \setminus S \). Because?
Lemma: Suppose that $S$ is a subset of nodes from $G = (V, E)$. Then, the cheapest edge $e$ between $S$ and $V \setminus S$ is part of every MST.

Proof: Any MST of $G$ must include some edge between $S$ and $V \setminus S$ (otherwise it would not be a tree).

Let $e$ be the cheapest edge between $S$ and $V \setminus S$.

Suppose $T$ is a ST that does not include $e$. Then:

1. $T \cup \{e\}$ must have a cycle. (Since spanning tree $T$ already has a path between $u$ and $v$, so adding $e$ will create a cycle)

2. That cycle must have another edge $e'$ between $S$ and $V \setminus S$. (Since there must be a path from $u \in S$ to $v \in V \setminus S$ in $T$)
**MST Cut Property**

**Lemma**: Suppose that $S$ is a subset of nodes from $G = (V, E)$. Then, the cheapest edge $e$ between $S$ and $V \setminus S$ is part of every MST.

**Proof**: Any MST of $G$ must include some edge between $S$ and $V \setminus S$ (otherwise it would not be a tree).

Let $e$ be the cheapest edge between $S$ and $V \setminus S$.

Suppose $T$ is a ST that does not include $e$. Then:

1. $T \cup \{e\}$ must have a cycle. (Since spanning tree $T$ already has a path between $u$ and $v$, so adding $e$ will create a cycle)

2. That cycle must have another edge $e'$ between $S$ and $V \setminus S$. (Since there must be a path from $u \in S$ to $v \in V \setminus S$ in $T$)
**MST Cut Property**

**Lemma:** Suppose that $S$ is a subset of nodes from $G = (V, E)$. Then, the cheapest edge $e$ between $S$ and $V \setminus S$ is part of every MST.

**Proof:** Any MST of $G$ must include some edge between $S$ and $V \setminus S$ (otherwise it would not be a tree).

Let $e$ be the cheapest edge between $S$ and $V \setminus S$.

Suppose $T$ is a ST that does not include $e$. Then:
1. $T \cup \{e\}$ must have a cycle. (Since spanning tree $T$ already has a path between $u$ and $v$, so adding $e$ will create a cycle)

2. That cycle must have another edge $e'$ between $S$ and $V \setminus S$. (Since there must be a path from $u \in S$ to $v \in V \setminus S$ in $T$)

**Need to make sure we pick the edge between $S$ and $V \setminus S$ on the cycle!**
Lemma: Suppose that $S$ is a subset of nodes from $G = (V, E)$. Then, the cheapest edge $e$ between $S$ and $V \setminus S$ is part of every MST.

Proof: Any MST of $G$ must include some edge between $S$ and $V \setminus S$ (otherwise it would not be a tree).

Let $e$ be the cheapest edge between $S$ and $V \setminus S$.

Suppose $T$ is a ST that does not include $e$. Then:

1. $T \cup \{e\}$ must have a cycle. (Since spanning tree $T$ already has a path between $u$ and $v$, so adding $e$ will create a cycle)

2. That cycle must have another edge $e'$ between $S$ and $V \setminus S$. (Since there must be a path from $u \in S$ to $v \in V \setminus S$ in $T$)

Need to make sure we pick the edge between $S$ and $V \setminus S$ on the cycle!
MST Cut Property

Lemma: Suppose that $S$ is a subset of nodes from $G = (V, E)$. Then, the cheapest edge $e$ between $S$ and $V \setminus S$ is part of every MST.

Proof: Any MST of $G$ must include some edge between $S$ and $V \setminus S$ (otherwise it would not be a tree).

Let $e$ be the cheapest edge between $S$ and $V \setminus S$.

Suppose $T$ is a ST that does not include $e$. $T \cup \{e\}$ must have a cycle and that cycle must have another edge $e'$ between $S$ and $V \setminus S$. 
Lemma: Suppose that $S$ is a subset of nodes from $G = (V, E)$. Then, the cheapest edge $e$ between $S$ and $V \setminus S$ is part of every MST.

Proof: Any MST of $G$ must include some edge between $S$ and $V \setminus S$ (otherwise it would not be a tree).

Let $e$ be the cheapest edge between $S$ and $V \setminus S$.

Suppose $T$ is a ST that does not include $e$. $T \cup \{e\}$ must have a cycle and that cycle must have another edge $e'$ between $S$ and $V \setminus S$.

Remove $e'$ to form $T' = T \cup \{e\} \setminus \{e'\}$.
**MST Cut Property**

**Lemma:** Suppose that $S$ is a subset of nodes from $G = (V, E)$. Then, the cheapest edge $e$ between $S$ and $V \setminus S$ is part of every MST.

**Proof:** Any MST of $G$ must include some edge between $S$ and $V \setminus S$ (otherwise it would not be a tree).

Let $e$ be the cheapest edge between $S$ and $V \setminus S$.

Suppose $T$ is a ST that does not include $e$. $T \cup \{e\}$ must have a cycle and that cycle must have another edge $e'$ between $S$ and $V \setminus S$.

Remove $e'$ to form $T' = T \cup \{e\} \backslash \{e'\}$.

Claim: $T'$ is a cheaper ST.
MST Cut Property

**Lemma**: Suppose that $S$ is a subset of nodes from $G = (V, E)$. Then, the cheapest edge $e$ between $S$ and $V \setminus S$ is part of every MST.

**Proof**: Any MST of $G$ must include some edge between $S$ and $V \setminus S$ (otherwise it would not be a tree).

Let $e$ be the cheapest edge between $S$ and $V \setminus S$.

Suppose $T$ is a ST that does not include $e$. $T \cup \{e\}$ must have a cycle and that cycle must have another edge $e'$ between $S$ and $V \setminus S$.

Remove $e'$ to form $T' = T \cup \{e\} \setminus \{e'\}$.

$T'$ is a tree (removing edge from cycle cannot disconnect graph)

$T'$ spans $V$ (same number of edges as ST $T$)
MST Cut Property

**Lemma:** Suppose that $S$ is a subset of nodes from $G = (V, E)$. Then, the cheapest edge $e$ between $S$ and $V \setminus S$ is part of every MST.

**Proof:** Any MST of $G$ must include some edge between $S$ and $V \setminus S$ (otherwise it would not be a tree).

Let $e$ be the cheapest edge between $S$ and $V \setminus S$.

Suppose $T$ is a ST that does not include $e$. $T \cup \{e\}$ must have a cycle and that cycle must have another edge $e'$ between $S$ and $V \setminus S$.

Remove $e'$ to form $T' = T \cup \{e\} \setminus \{e'\}$.

$T'$ is a tree (removing edge from cycle cannot disconnect graph)

$T'$ spans $V$ (same number of edges as ST $T$)

$\text{weight}(T') = \text{weight}(T) + \text{weight}(e) - \text{weight}(e')$. 

$$
\begin{align*}
\text{weight}(T') &= \text{weight}(T) + \text{weight}(e) - \text{weight}(e') \\
\end{align*}
$$
Lemma: Suppose that $S$ is a subset of nodes from $G = (V, E)$. Then, the cheapest edge $e$ between $S$ and $V \setminus S$ is part of every MST.

Proof: Any MST of $G$ must include some edge between $S$ and $V \setminus S$ (otherwise it would not be a tree).

Let $e$ be the cheapest edge between $S$ and $V \setminus S$.

Suppose $T$ is a ST that does not include $e$. $T \cup \{e\}$ must have a cycle and that cycle must have another edge $e'$ between $S$ and $V \setminus S$.

Remove $e'$ to form $T' = T \cup \{e\} \setminus \{e'\}$.

$T'$ is a tree (removing edge from cycle cannot disconnect graph) $T'$ spans $V$ (same number of edges as ST $T$)

weight($T'$) = weight($T$) + weight($e$) − weight($e'$).

$\Rightarrow$ weight($T'$) < weight($T$), since weight($e$) < weight($e'$).
**MST Cut Property**

**Lemma:** Suppose that $S$ is a subset of nodes from $G = (V, E)$. Then, the cheapest edge $e$ between $S$ and $V \setminus S$ is part of every MST.

**Proof:** Any MST of $G$ must include some edge between $S$ and $V \setminus S$ (otherwise it would not be a tree).

Let $e$ be the cheapest edge between $S$ and $V \setminus S$.

Suppose $T$ is a ST that does not include $e$. $T \cup \{e\}$ must have a cycle and that cycle must have another edge $e'$ between $S$ and $V \setminus S$.

Remove $e'$ to form $T' = T \cup \{e\} \setminus \{e'\}$.

$T'$ is a tree (removing edge from cycle cannot disconnect graph)

$T'$ spans $V$ (same number of edges as ST $T$)

$weight(T') = weight(T) + weight(e) - weight(e')$.

$\Rightarrow weight(T') < weight(T)$, since $weight(e) < weight(e')$.

$\Rightarrow T'$ is a cheaper ST.
**MST Cut Property**

**Lemma:** Suppose that $S$ is a subset of nodes from $G = (V, E)$. Then, the cheapest edge $e$ between $S$ and $V \setminus S$ is part of every MST.

**Proof:** Any MST of $G$ must include some edge between $S$ and $V \setminus S$ (otherwise it would not be a tree).

Let $e$ be the cheapest edge between $S$ and $V \setminus S$.

Suppose $T$ is a ST that does not include $e$. $T \cup \{e\}$ must have a cycle and that cycle must have another edge $e'$ between $S$ and $V \setminus S$.

Remove $e'$ to form $T' = T \cup \{e\} \setminus \{e'\}$.

$T'$ is a tree (removing edge from cycle cannot disconnect graph)

$T'$ spans $V$ (same number of edges as ST $T$)

weight($T'$) = weight($T$) + weight($e$) − weight($e'$).

$\Rightarrow$ weight($T'$) < weight($T$), since weight($e$) < weight($e'$).

$\Rightarrow$ $T'$ is a cheaper ST.

So, $e$ is part of every MST.