4. Greedy Algorithms II

- red-rule blue-rule demo
- Prim’s algorithm demo
- Kruskal’s algorithm demo
- Boruvka’s algorithm demo
4. **Greedy Algorithms II**

- red-rule blue-rule demo
- Prim’s algorithm demo
- Kruskal’s algorithm demo
- Boruvka’s algorithm demo
Red-rule blue-rule demo

Red rule. Let $C$ be a cycle with no red edges. Select an uncolored edge of $C$ of max weight and color it red.

Blue rule. Let $D$ be a cutset with no blue edges. Select an uncolored edge in $D$ of min weight and color it blue.

the input graph

![Diagram of a graph with nodes and edges labeled with weights from 1 to 9. The edges include: 1-5, 5-7, 7-1, 1-9, 9-2, 2-4, 4-6, 6-7, 7-3, 3-8, 8-2.]
Red-rule blue-rule demo

Red rule. Let $C$ be a cycle with no red edges. Select an uncolored edge of $C$ of max weight and color it red.

apply the red rule to the cycle
Red-rule blue-rule demo

current set of red and blue edges
Red-rule blue-rule demo

**Red rule.** Let $C$ be a cycle with no red edges. Select an uncolored edge of $C$ of max weight and color it red.

apply the red rule to the cycle
Red-rule blue-rule demo

Red rule. Let $C$ be a cycle with no red edges. Select an uncolored edge of $C$ of max weight and color it red.

current set of red and blue edges

![Diagram showing a cycle with edges colored red and blue. The red edges are highlighted in brown.]
Blue rule. Let $D$ be a cutset with no blue edges. Select an uncolored edge in $D$ of min weight and color it blue.
Red-rule blue-rule demo

current set of red and blue edges
Red-rule blue-rule demo

Blue rule. Let $D$ be a cutset with no blue edges. Select an uncolored edge in $D$ of min weight and color it blue.

apply the blue rule to the cutset
Red-rule blue-rule demo

current set of red and blue edges
Red rule. Let $C$ be a cycle with no red edges. Select an uncolored edge of $C$ of max weight and color it red.

apply the red rule to the cycle
Red-rule blue-rule demo

current set of red and blue edges
Red-rule blue-rule demo

**Blue rule.** Let $D$ be a cutset with no blue edges. Select an uncolored edge in $D$ of min weight and color it blue.

apply the blue rule to the cutset
Red-rule blue-rule demo

current set of red and blue edges
Red-rule blue-rule demo

**Blue rule.** Let $D$ be a cutset with no blue edges. Select an uncolored edge in $D$ of min weight and color it blue.

apply the blue rule to the cutset
Red-rule blue-rule demo

current set of red and blue edges
**Blue rule.** Let $D$ be a cutset with no blue edges. Select an uncolored edge in $D$ of min weight and color it blue.

apply the blue rule to the cutset
Red-rule blue-rule demo

current set of red and blue edges
**Blue rule.** Let $D$ be a cutset with no blue edges. Select an uncolored edge in $D$ of min weight and color it blue.
Red-rule blue-rule demo

current set of red and blue edges
Red-rule blue-rule demo

**Greedy algorithm.** Upon termination, the blue edges form a MST.

*a minimum spanning tree*
4. Greedy Algorithms II

- red-rule blue-rule demo
- Prim’s algorithm demo
- Kruskal’s algorithm demo
- Boruvka’s algorithm demo

Section 4.5
Prim’s algorithm demo

Initialize $S = $ any node, $T = \emptyset$.
Repeat $n – 1$ times:
  • Add to $T$ a min-weight edge with one endpoint in $S$.
  • Add new node to $S$. 
Prim’s algorithm demo

Initialize $S =$ any node, $T = \emptyset$.
Repeat $n - 1$ times:
- Add to $T$ a min-weight edge with one endpoint in $S$.
- Add new node to $S$. 
Prim’s algorithm demo

Initialize $S = \text{any node}$, $T = \emptyset$.
Repeat $n - 1$ times:
  • Add to $T$ a min-weight edge with one endpoint in $S$.
  • Add new node to $S$. 

![Graph with Prim's algorithm example]
Prim’s algorithm demo

Initialize $S = \text{any node}$, $T = \emptyset$.
Repeat $n - 1$ times:
  • Add to $T$ a min-weight edge with one endpoint in $S$.
  • Add new node to $S$. 

![Diagram of Prim's algorithm](attachment:image.png)
Prim’s algorithm demo

Initialize $S$ = any node, $T = \emptyset$.
Repeat $n - 1$ times:
  • Add to $T$ a min-weight edge with one endpoint in $S$.
  • Add new node to $S$. 
Prim’s algorithm demo

Initialize $S =$ any node, $T = \emptyset$.
Repeat $n - 1$ times:
  • Add to $T$ a min-weight edge with one endpoint in $S$.
  • Add new node to $S$. 

Prim’s algorithm demo

Initialize $S =$ any node, $T =$ $\emptyset$.
Repeat $n - 1$ times:
- Add to $T$ a min-weight edge with one endpoint in $S$.
- Add new node to $S$.
Prim’s algorithm demo

Initialize $S = \text{any node}$, $T = \emptyset$.
Repeat $n - 1$ times:
  • Add to $T$ a min-weight edge with one endpoint in $S$.
  • Add new node to $S$. 

![Graph with weights](image-url)
Prim’s algorithm demo

Initialize $S = \text{any node}$, $T = \emptyset$.

Repeat $n - 1$ times:

- Add to $T$ a min-weight edge with one endpoint in $S$.
- Add new node to $S$. 
Prim’s algorithm demo

Initialize $S = \text{any node}$, $T = \emptyset$.
Repeat $n - 1$ times:
  • Add to $T$ a min-weight edge with one endpoint in $S$.
  • Add new node to $S$. 
Prim’s algorithm demo

Initialize $S = $ any node, $T = \emptyset$.
Repeat $n - 1$ times:
  • Add to $T$ a min-weight edge with one endpoint in $S$.
  • Add new node to $S$. 
**Prim’s algorithm demo**

Initialize $S = \text{any node}$, $T = \emptyset$.

Repeat $n - 1$ times:
- Add to $T$ a min-weight edge with one endpoint in $S$.
- Add new node to $S$. 

![Diagram of Prim's algorithm demonstration](image)
Prim’s algorithm demo

Initialize $S = \text{any node}$, $T = \emptyset$.

Repeat $n - 1$ times:
- Add to $T$ a min-weight edge with one endpoint in $S$.
- Add new node to $S$. 
Prim’s algorithm demo

Initialize $S = \text{any node}$, $T = \emptyset$.
Repeat $n - 1$ times:
  - Add to $T$ a min-weight edge with one endpoint in $S$.
  - Add new node to $S$. 

![Graph showing Prim's algorithm demonstration](image-url)
Prim’s algorithm demo

Initialize $S = \text{any node}$, $T = \emptyset$.
Repeat $n - 1$ times:
  • Add to $T$ a min-weight edge with one endpoint in $S$.
  • Add new node to $S$. 

![Diagram of Prim’s algorithm](image-url)
Prim’s algorithm demo

Initialize \( S = \text{any node}, \ T = \emptyset \).
Repeat \( n - 1 \) times:
  - Add to \( T \) a min-weight edge with one endpoint in \( S \).
  - Add new node to \( S \).
Prim’s algorithm demo

Initialize \( S = \) any node, \( T = \emptyset \).

Repeat \( n - 1 \) times:

- Add to \( T \) a min-weight edge with one endpoint in \( S \).
- Add new node to \( S \).
4. Greedy Algorithms II

- red-rule blue-rule demo
- Prim’s algorithm demo
- Kruskal’s algorithm demo
- Boruvka’s algorithm demo

Section 4.5
Kruskal’s algorithm demo

Consider edges in ascending order of weight:
- Add to \( T \) unless it would create a cycle.
Kruskal’s algorithm demo

Consider edges in ascending order of weight:
- Add to $T$ unless it would create a cycle.
Kruskal’s algorithm demo

Consider edges in ascending order of weight:
- Add to $T$ unless it would create a cycle.
Kruskal’s algorithm demo

Consider edges in ascending order of weight:
- Add to $T$ unless it would create a cycle.
Kruskal’s algorithm demo

Consider edges in ascending order of weight:
  • Add to $T$ unless it would create a cycle.
Kruskal’s algorithm demo

Consider edges in ascending order of weight:
  • Add to $T$ unless it would create a cycle.
Kruskal’s algorithm demo

Consider edges in ascending order of weight:

- Add to $T$ unless it would create a cycle.
Kruskal’s algorithm demo

Consider edges in ascending order of weight:
  • Add to $T$ unless it would create a cycle.
Kruskal’s algorithm demo

Consider edges in ascending order of weight:
• Add to $T$ unless it would create a cycle.
Kruskal’s algorithm demo

Consider edges in ascending order of weight:

- Add to $T$ unless it would create a cycle.
Kruskal’s algorithm demo

Consider edges in ascending order of weight:
- Add to $T$ unless it would create a cycle.
Kruskal’s algorithm demo

Consider edges in ascending order of weight:

- Add to $T$ unless it would create a cycle.
4. **Greedy Algorithms II**

- red-rule blue-rule demo
- Prim’s algorithm demo
- Kruskal’s algorithm demo
- Boruvka’s algorithm demo
Borůvka’s algorithm demo

Repeat until only one tree.

- Apply blue rule to cutset corresponding to each blue tree.
- Color all selected edges blue.
Borůvka’s algorithm demo

Repeat until only one tree.
- Apply blue rule to cutset corresponding to each blue tree.
- Color all selected edges blue.
Borůvka’s algorithm demo

Repeat until only one tree.

- Apply blue rule to cutset corresponding to each blue tree.
- Color all selected edges blue.
Borůvka’s algorithm demo

Repeat until only one tree.

• Apply blue rule to cutset corresponding to each blue tree.
• Color all selected edges blue.
Borůvka’s algorithm demo

Repeat until only one tree.

- Apply blue rule to cutset corresponding to each blue tree.
- Color all selected edges blue.
Borůvka’s algorithm demo

Repeat until only one tree.
  • Apply blue rule to cutset corresponding to each blue tree.
  • Color all selected edges blue.
Borůvka’s algorithm demo

Repeat until only one tree.

- Apply blue rule to cutset corresponding to each blue tree.
- Color all selected edges blue.