Data Structures and Algorithms

(CS210A)

Semester I – **2014-15**

Lecture 36

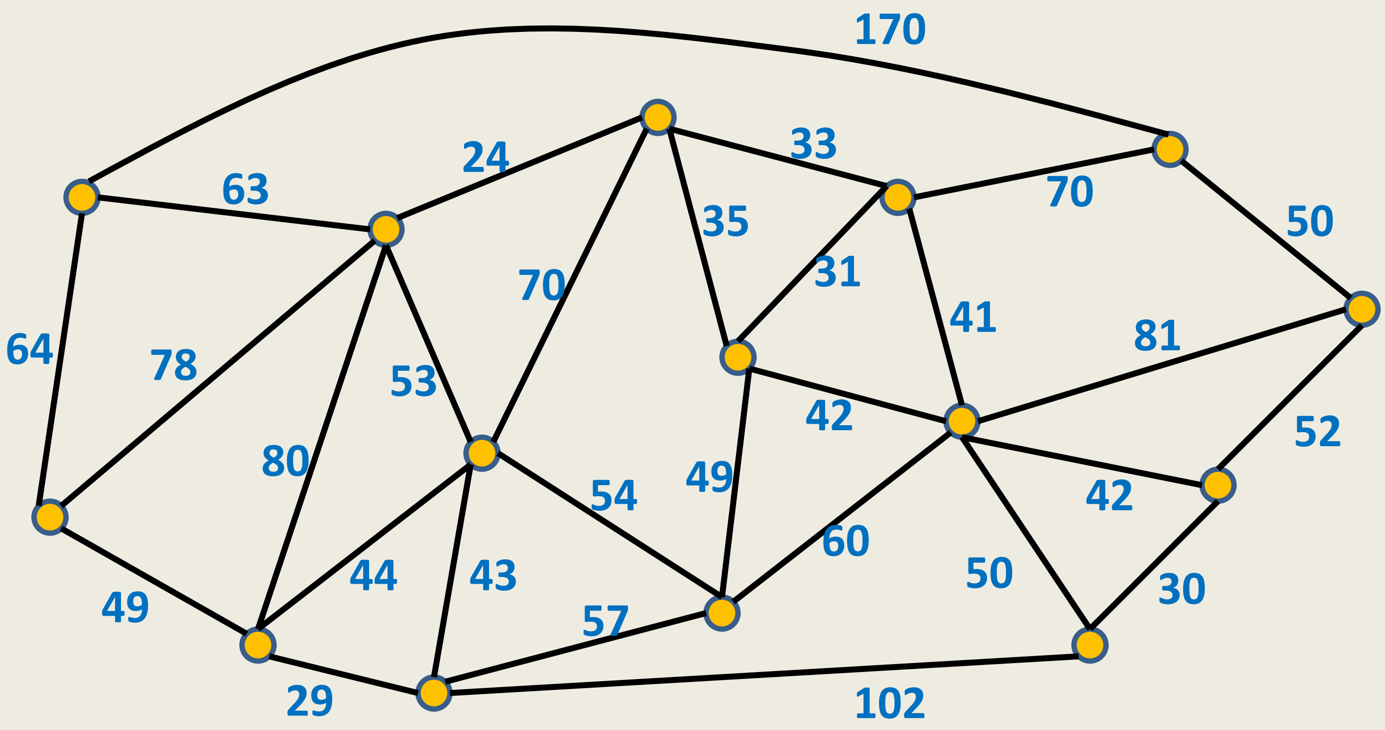
- **A new algorithm design paradigm:** Greedy strategy

part III

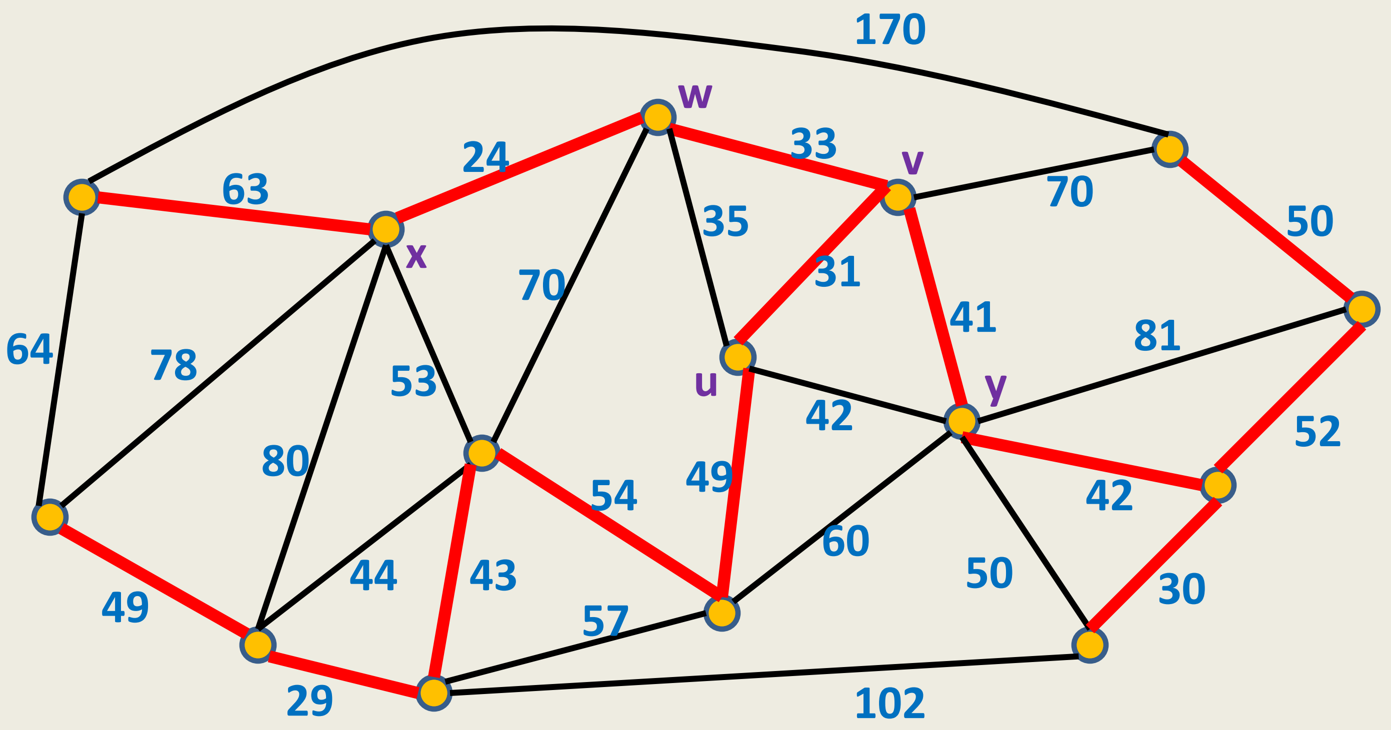
Continuing Problem from last class

Minimum spanning tree

Minimum Spanning Tree (MST)



Minimum Spanning Tree (MST)



Problem Description

Input: an undirected graph G=(V,E) with $\mathbf{w}$: E → $\mathbb{R}$,

Aim: compute a **spanning tree** (V, E'), $E' \subseteq E$ such that $\sum_{e \in E'} \mathbf{w}(e)$ is **minimum.**

Lemma (proved in last class):

If $e_0 \in E$ is the edge of **least weight** in G, then there is a **MST** T containing e_0.

Homework:
Try once more to answer this question ☺.

A useful lesson for design of a graph algorithm

If you have a complicated algorithm for a graph problem, …

- search for **some graph theoretic property**

to design **simpler** and **more efficient** algorithm

Two graph theoretic properties of MST

- **Cut property**

- **Cycle property**

Every algorithm till date is **based on** one of **these properties**!

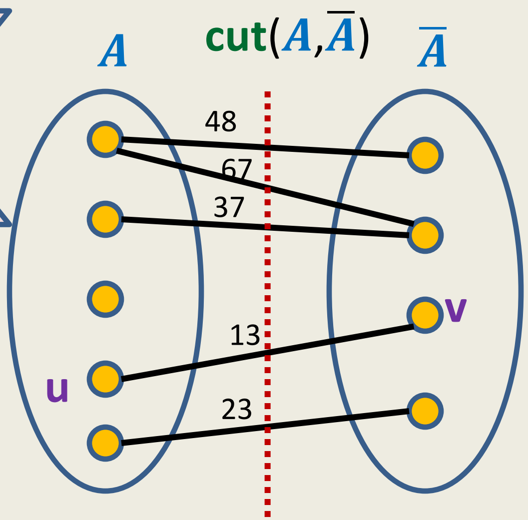
Cut Property

Cut Property

Definition: For any subset $A \subseteq V$, such that $\emptyset \neq A \neq V$,

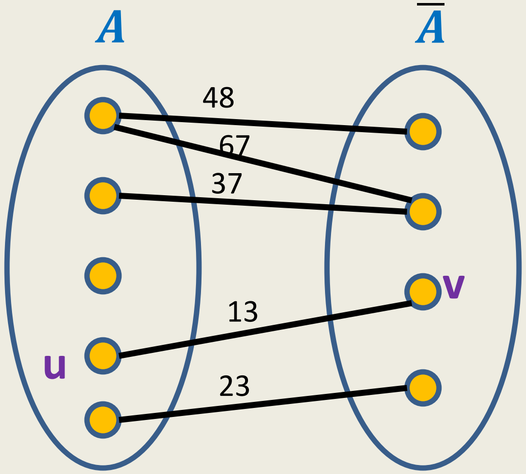
cut$(A,\overline{A})$ = { $(u,v) \in E$ | $u \in A$ **and** $v \in \overline{A}$ **or** $v \in A$ **and** $u \in \overline{A}$ }

Pursuing **greedy strategy** to minimize weight of MST, what can we say about the edges of **cut**$(A, \overline{A})$?



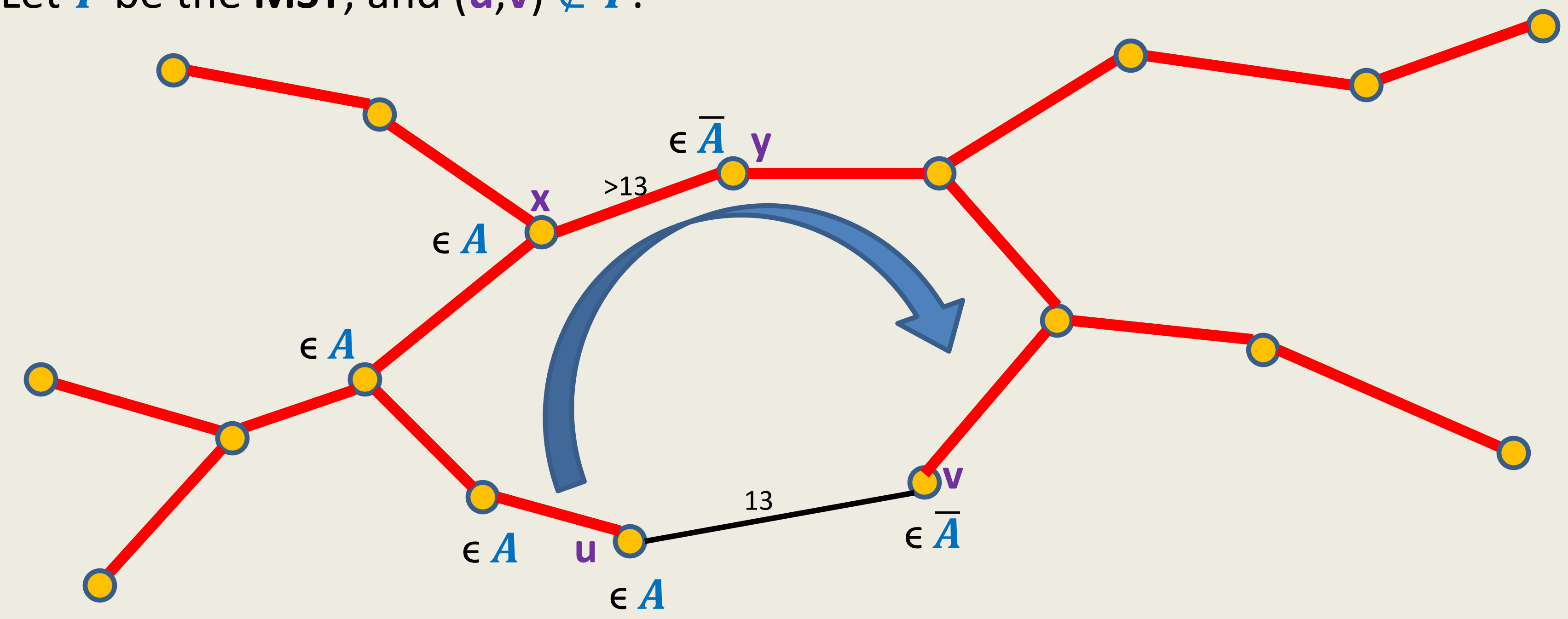
Cut-property: The **least weight edge** of a **cut**$(A,\overline{A})$ must be in **MST.**

Proof of cut-property



Proof of cut-property

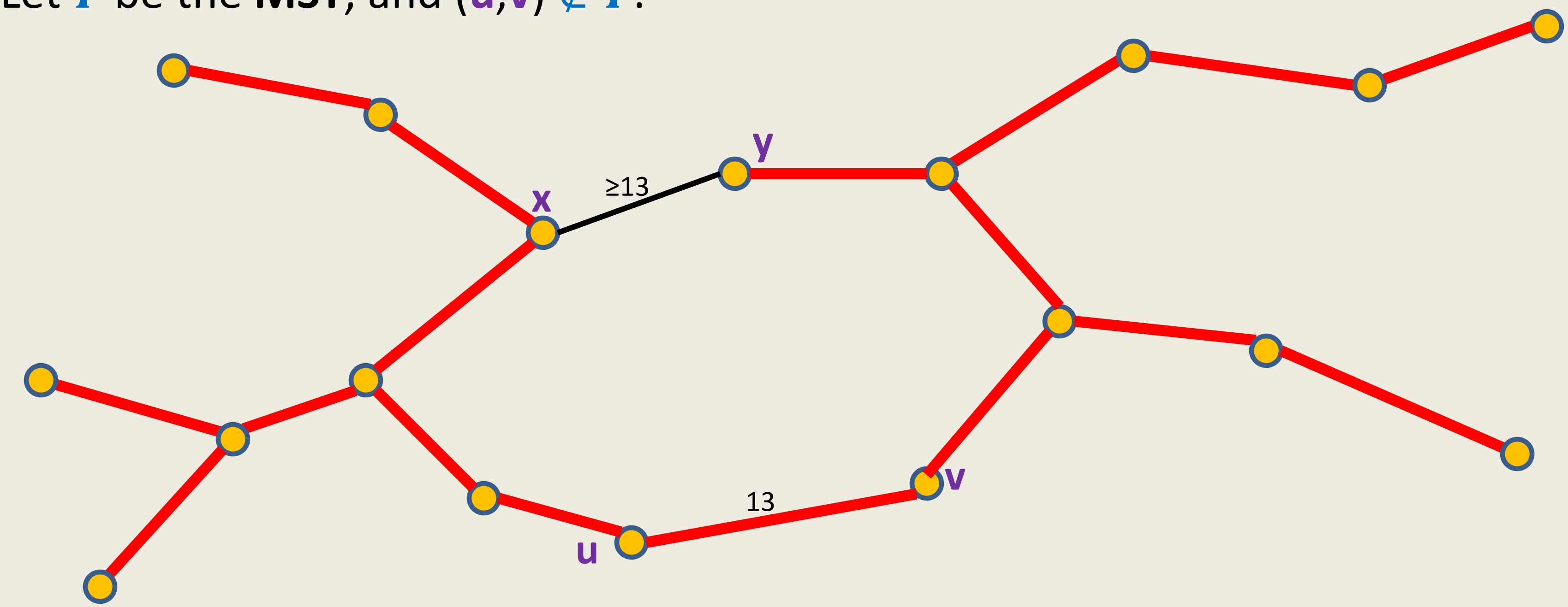
Let T be the **MST**, and (u,v) $\notin T$.



Question: What happens if we **remove** (x,y) from T, and **add** (u,v) to T.

Proof of cut-property

Let T be the **MST**, and (u,v) $\notin T$.



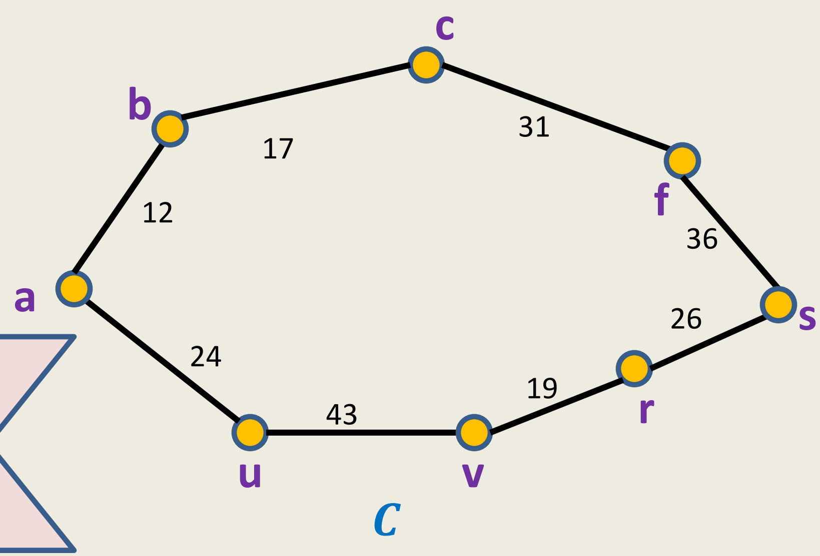
Question: What happens if we **remove** (x,y) from T, and **add** (u,v) to T.

We get a spanning tree T' with weight < **weight**(T)
A contradiction !

Cycle Property

Cycle Property

Let C be any cycle in G.

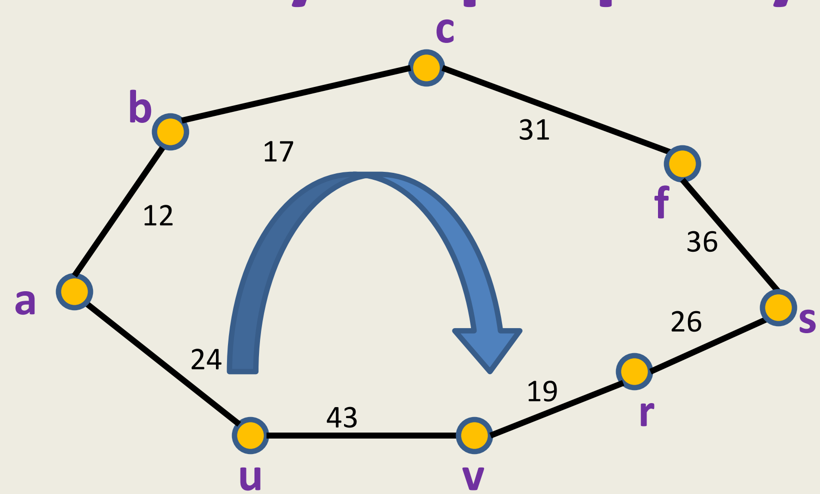


Pursuing **greedy strategy** to minimize weight of MST, what can we say about the edges of cycle C ?

Cycle-property:

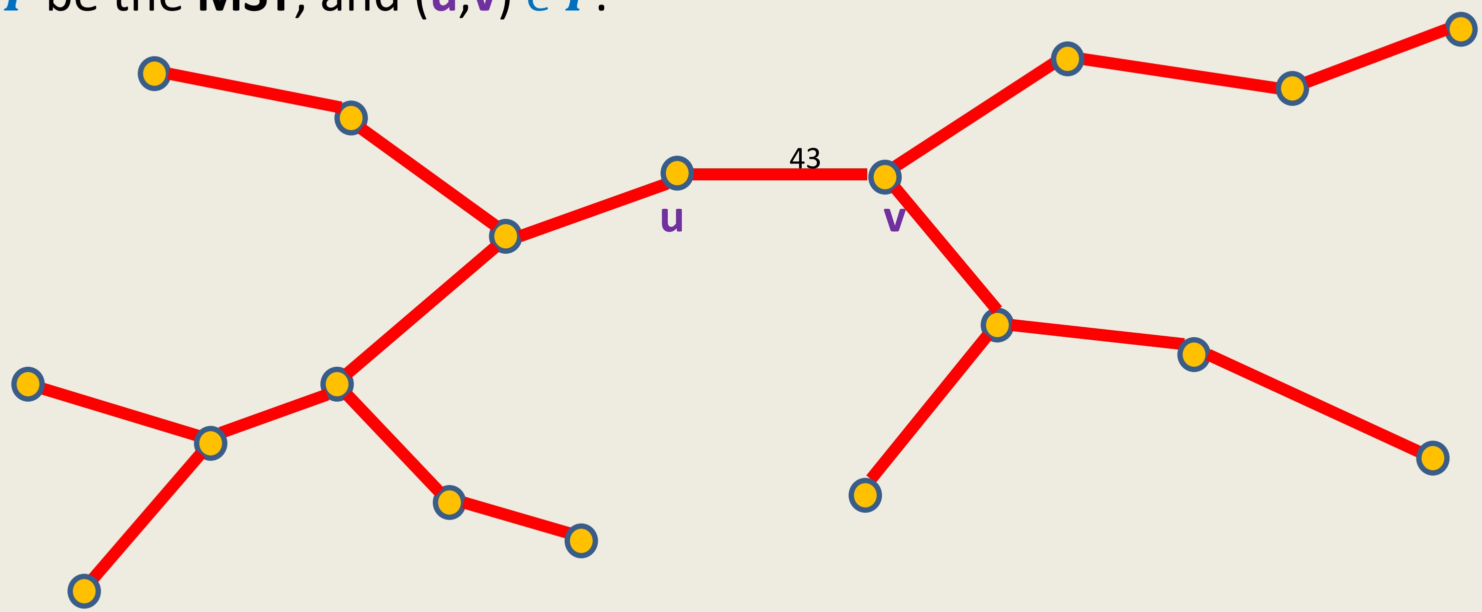
Maximum weight edge of any cycle C **can not** be present in **MST**.

Proof of Cycle property



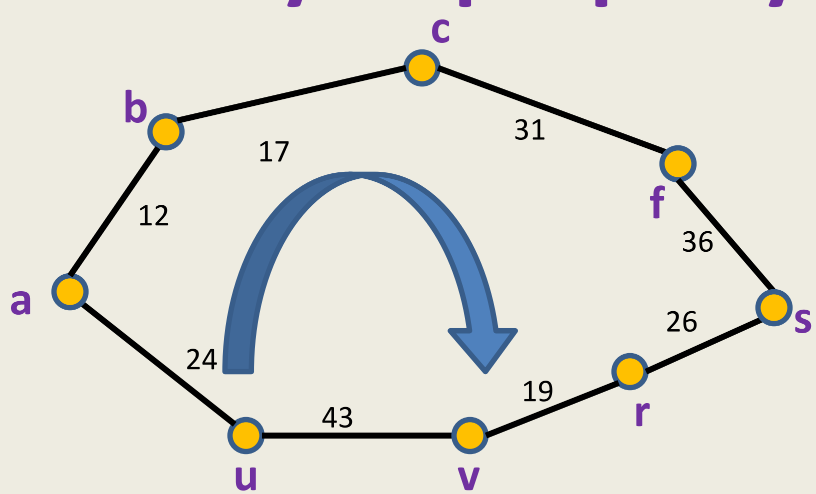
C

Let T be the **MST**, and (**u**,**v**) $\in T$.

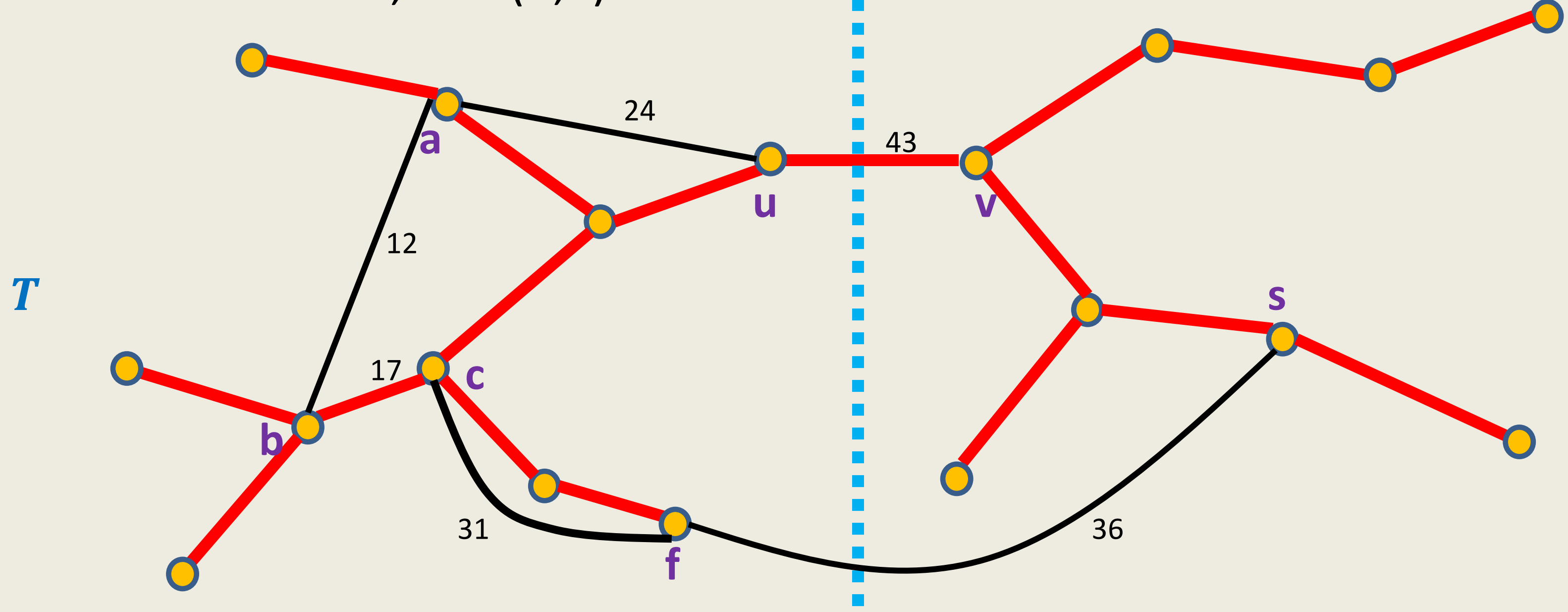


T

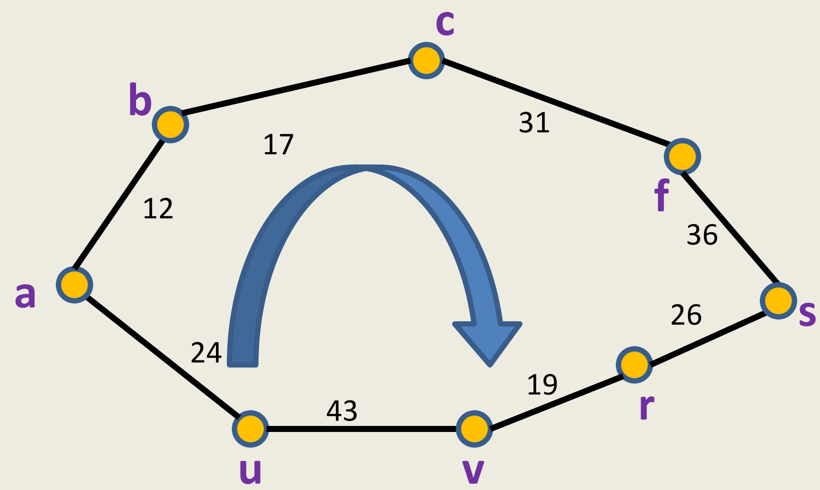
Proof of Cycle property



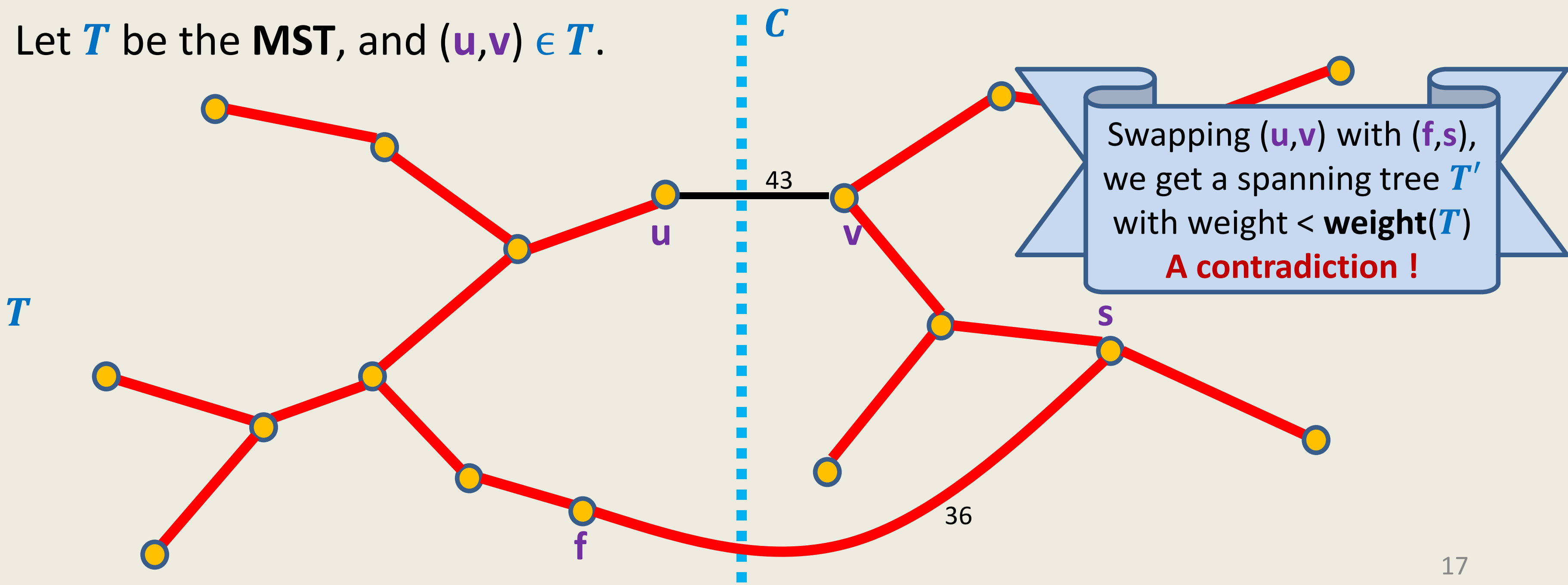
Let T be the **MST**, and (u,v) ϵ T.



Proof of Cycle property

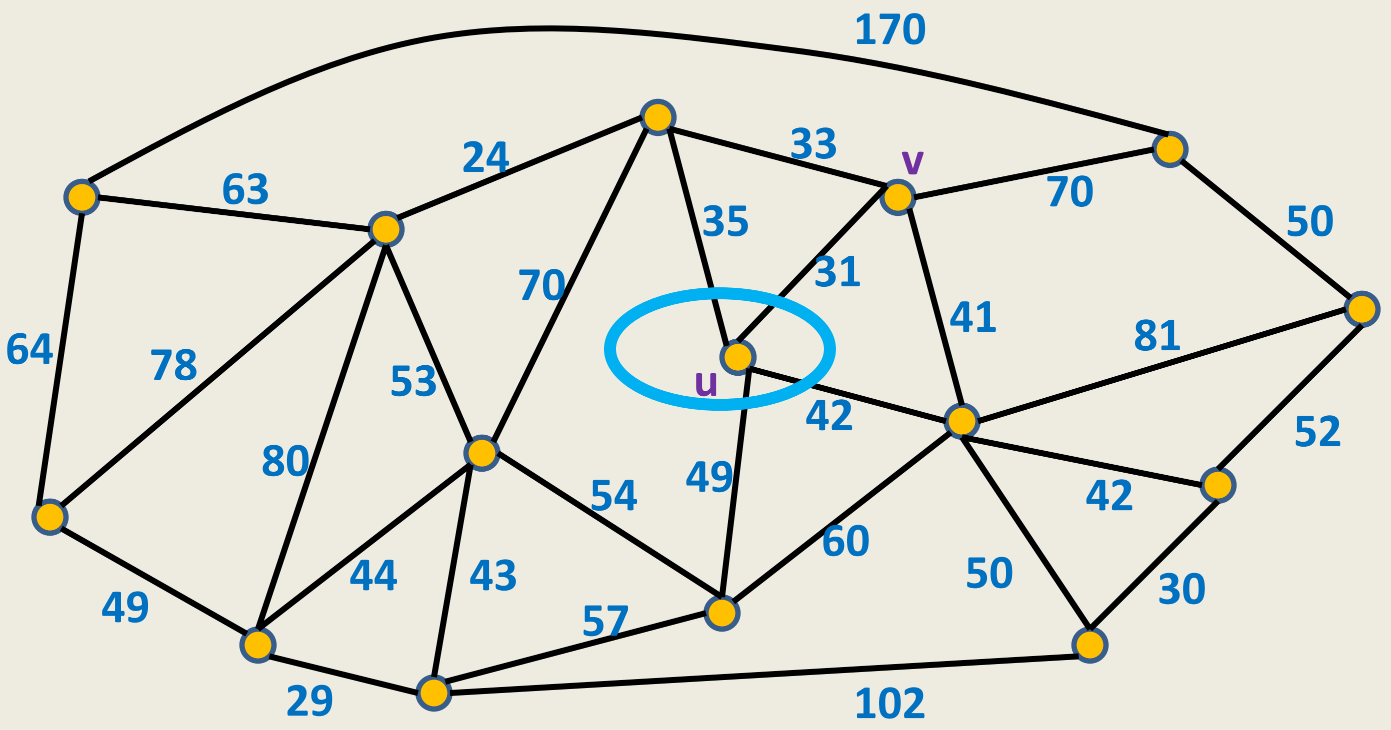


Let T be the **MST**, and (u,v) ∊ T.

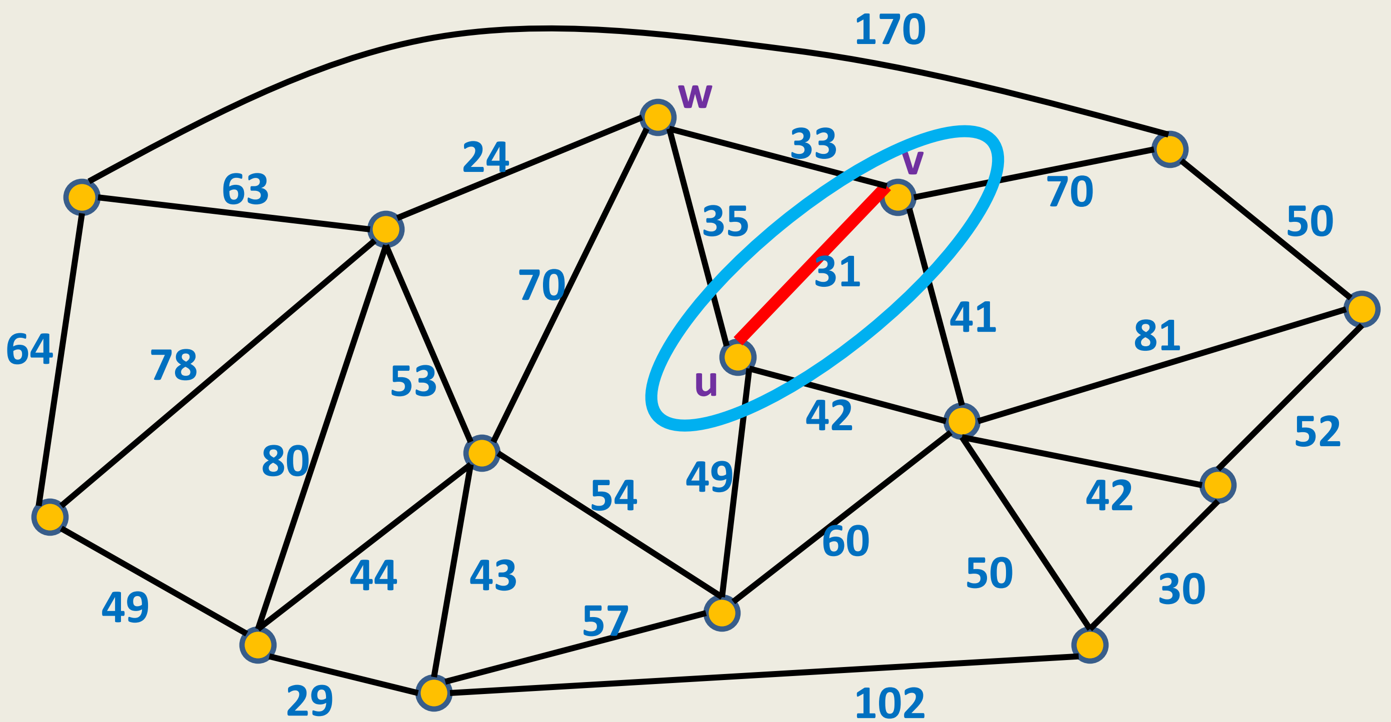


Algorithms based on cut Property

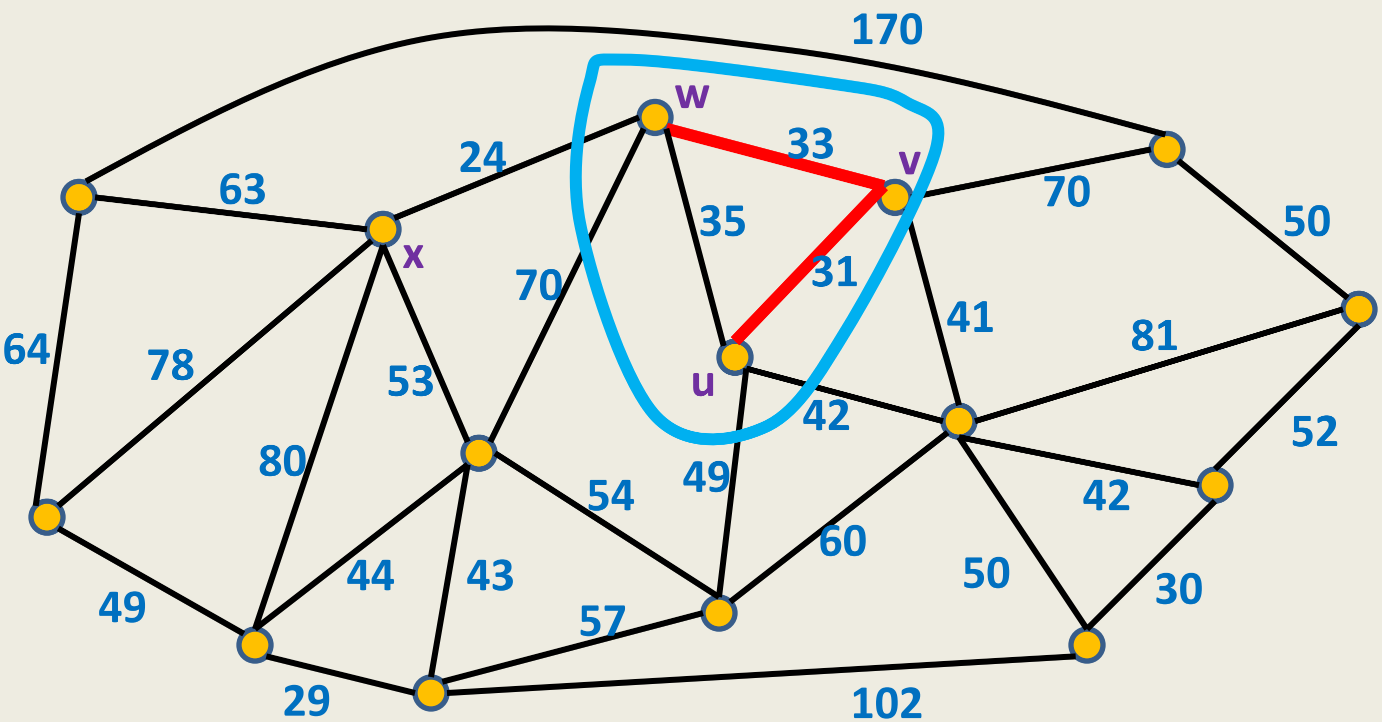
How to use cut property to compute a MST ?



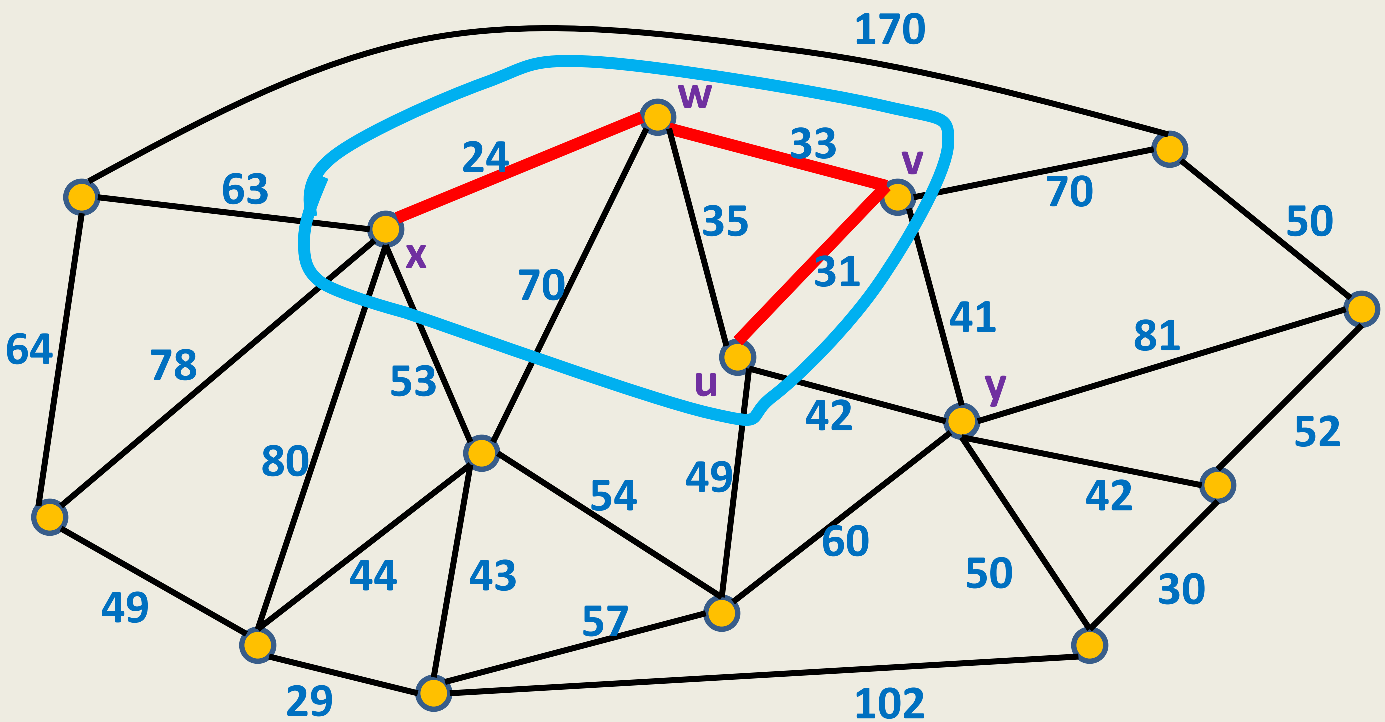
How to use cut property to compute a MST ?



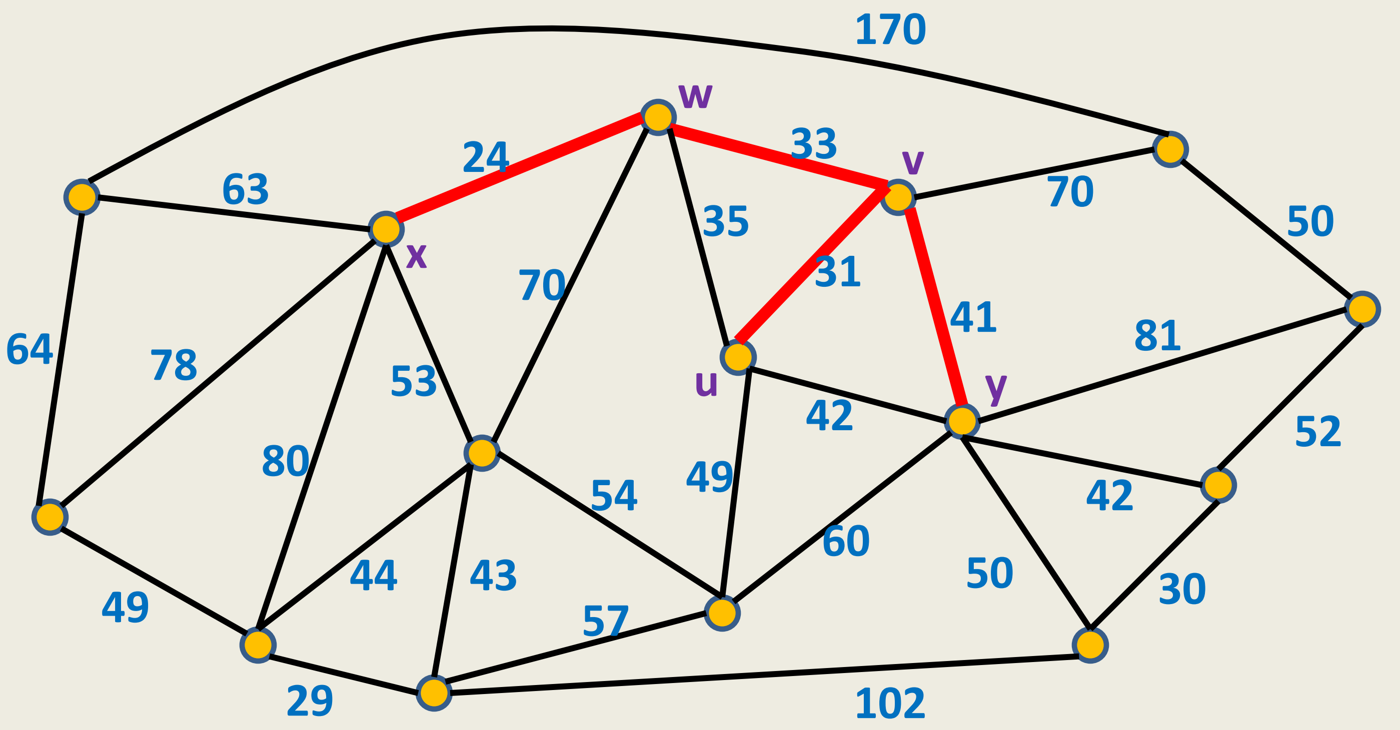
How to use cut property to compute a MST ?



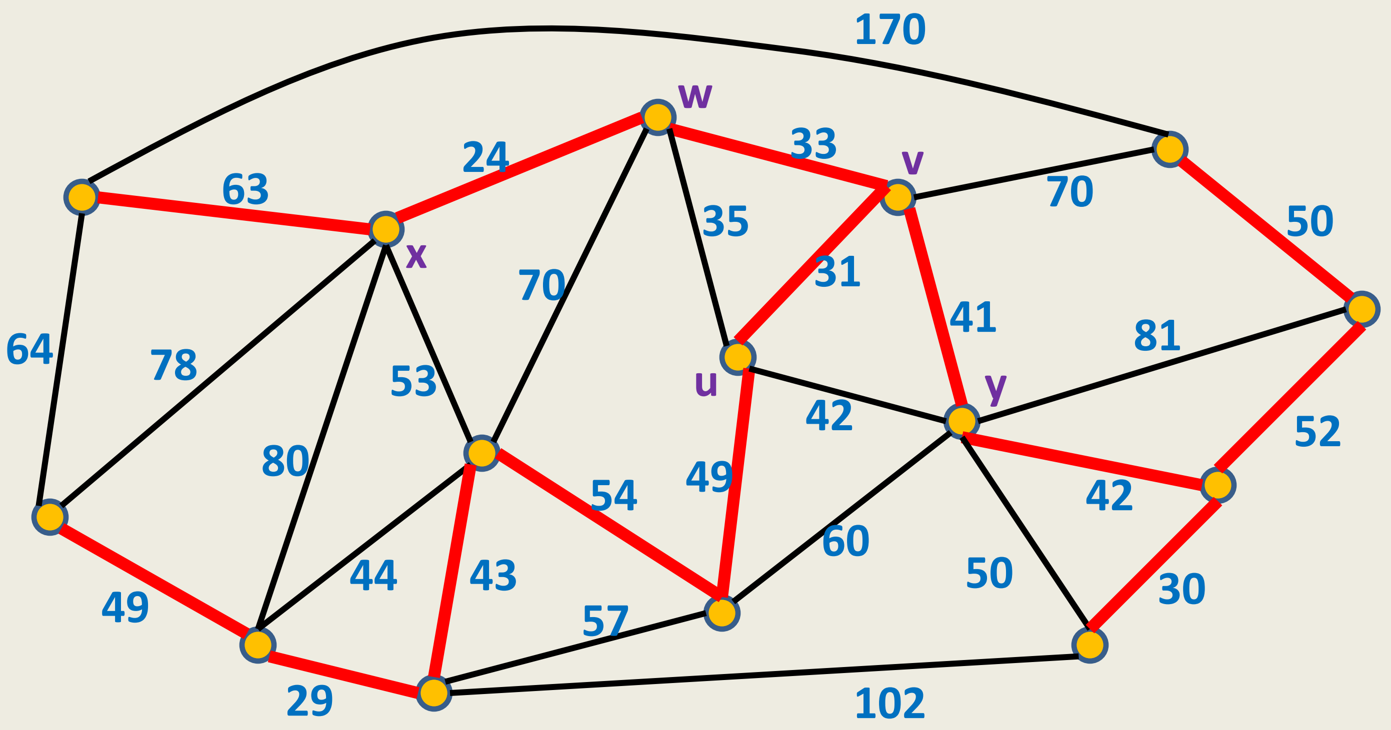
How to use cut property to compute a MST ?



How to use cut property to compute a MST ?



How to use cut property to compute a MST ?



An Algorithm based on cut property

Algorithm (Input: **graph** G =(V,E) with **weights** on edges)

T ← ∅;

A ← {u};

While ($A <> V$ **) do**

{ Compute the least weight edge from **cut**(A,$\bar{A}$);

Let this edge be (**x**,**y**), with **x**∊ A, **y**∊ $\bar{A}$;

T ← T ∪ {(**x**, **y**)};

A ← A ∪{**y**};

}

Return T;

Number of iterations of the **While** loop **:** $n - 1$

Time spent in one iteration of While loop: **O**(m)

➔ **Running time** of the algorithm: **O**(mn)

Algorithm based on cycle Property

An Algorithm based on cycle property

Description

Algorithm (Input: **graph** G =(V,E) with **weights** on edges)

While (E **has any cycle) do**

{ Compute any cycle C;

Let (**u**,**v**) be the **maximum weight** edge of the cycle C;

Remove (**u**,**v**) from E;

}

Return E;

Number of iterations of the **While** loop **:** $m - n + 1$

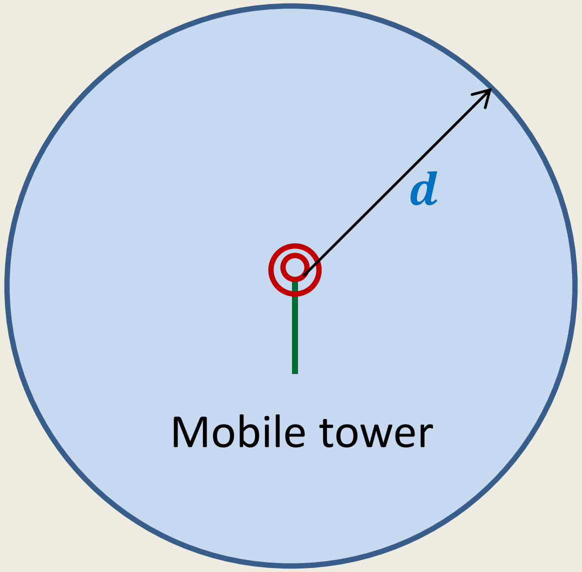
Time spent in one iteration of While loop: O(n)

➔ **Running time** of the algorithm: O(mn)

Problem 3

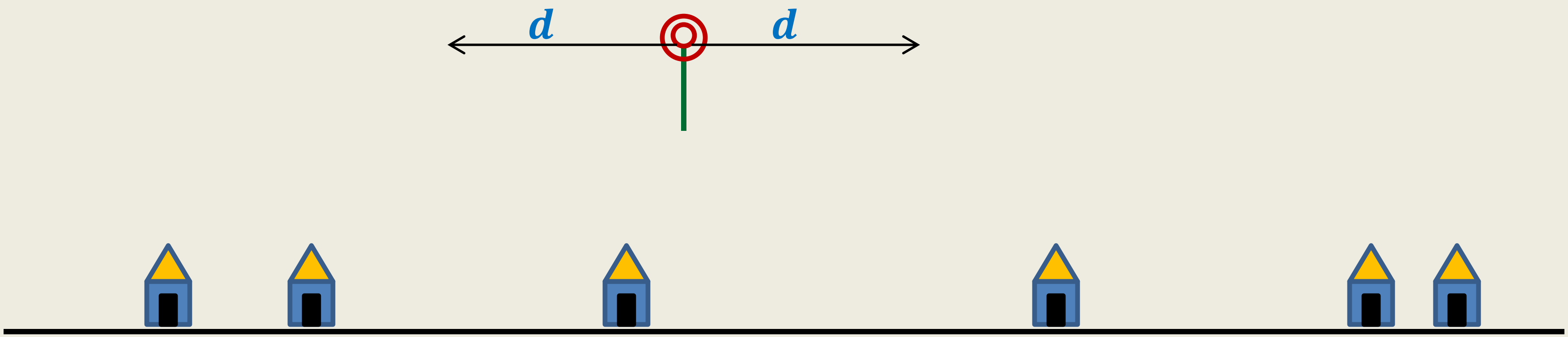
Mobile towers on a road

Mobile towers on a road



A mobile tower can cover any cell phone within radius d.

Mobile towers on a road



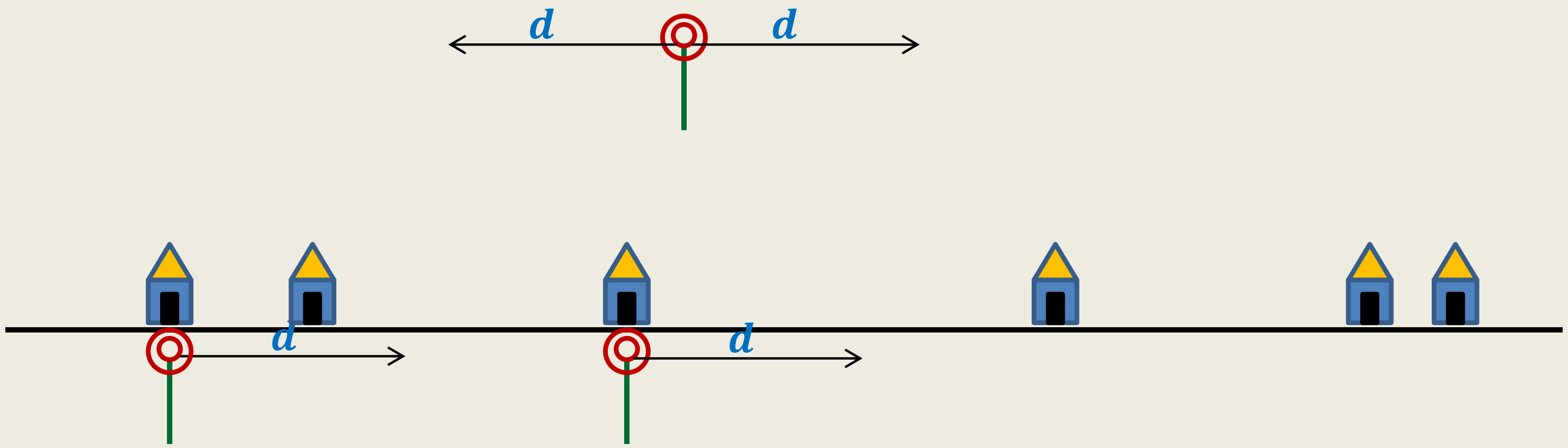
Problem statement:

There are n houses located along a road.

We want to place mobile towers such that

- Each house is **covered** by at least one mobile tower.
- The number of mobile towers used is **least** possible.

Mobile towers on a road



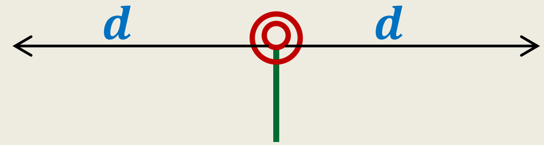
Strategy 1:

Place tower at first house,

Remove all houses covered by this tower.

Proceed to the next uncovered house …

Mobile towers on a road



can we say anything about the optimal solution ?

Strategy 2:

Place tower at distance d to the right of the first house;

Remove all houses covered by this tower;

Proceed to the next uncovered house along the road…

Lemma: There is an optimal solution for the problem in which the <u>leftmost</u> tower is placed at distance d <u>to the right</u> of the first house

Problem 4

Overlapping Intervals

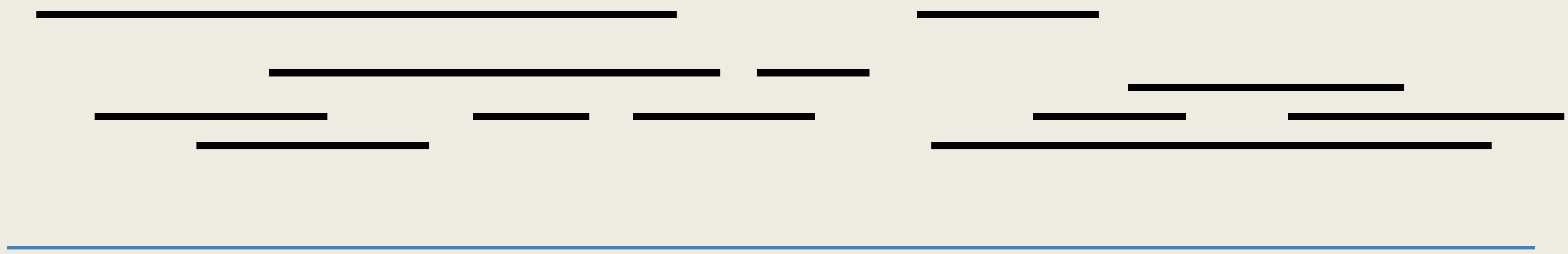
If you have started feeling that design of greedy algorithm is easy, this problem is going to prove you wrong ...☺

Overlapping Intervals

Problem statement:

Given a set **A** of n intervals, compute smallest set **B** of intervals so that for every interval **I** in **A\B**, there is some interval in **B** which overlaps/intersects with **I**.

A

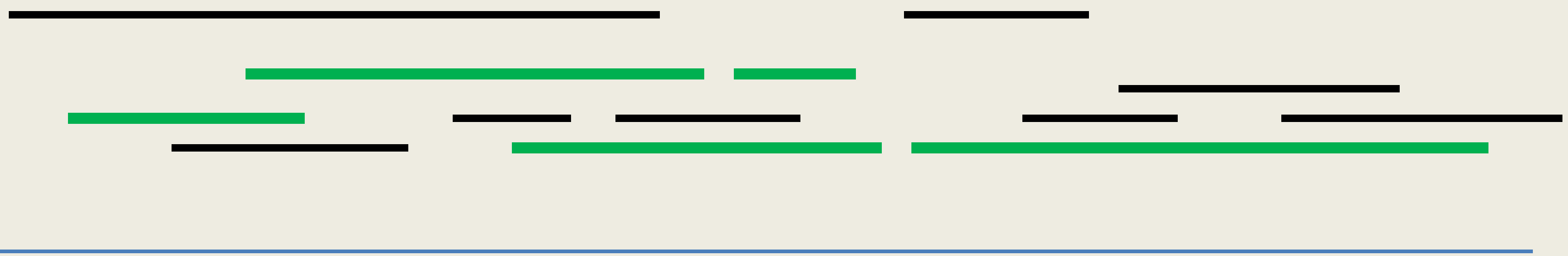


Overlapping Intervals

Problem statement:

Given a set **A** of n intervals, compute smallest set **B** of intervals so that for every interval **I** in **A\B**, there is some interval in **B** which overlaps/intersects with **I**.

A



another optimal solution ☺

A difficult problem ☹
We shall solve it in
one full lecture.

Homework …

Ponder over the following questions before coming for the next class

- Use **cycle property** and/or **cut property** to design a **new algorithm for MST**
- Use some data structure to improve the running time of the algorithms discussed in this class to **O**($\boldsymbol{m} \log \boldsymbol{n}$)