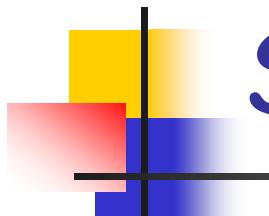


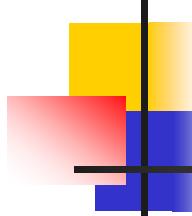
The Stackelberg Minimum Spanning Tree Game

J. Cardinal, E. Demaine, S. Fiorini, G. Joret, S. Langerman, I. Newman, O. Weimann,
[The Stackelberg Minimum Spanning Tree Game](#), WADS'07



Stackelberg Game

- 2 players: **leader** and **follower**
- The leader moves first, then the follower moves
- The follower optimizes his objective function
 - ...knowing the leader's move
- The leader optimizes his objective function
 - ...by anticipating the optimal response of the follower
- **Our goal:** to find a good strategy for the leader

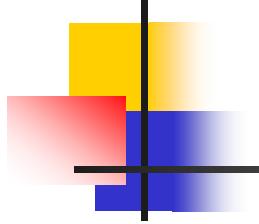


Setting

- We have a graph $G=(V,E)$, with $E=R \cup B$
- each $e \in R$, has a fixed positive cost $c(e)$
- Leader owns B , and has to set a price $p(e)$ for each $e \in B$
- function c and function p define a weight function $w: E \rightarrow \mathbb{R}^+$
- the follower buys an MST T of G (w.r.t. to w)
- Leader's revenue of T is:

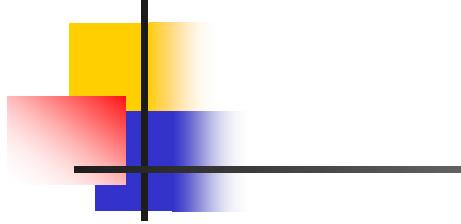
$$\sum_{e \in E(T) \cap B} p(e)$$

goal: find prices in order to maximize the revenue



There is a trade-off:

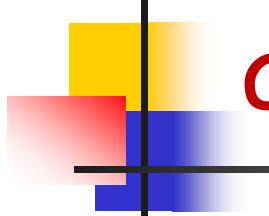
- Leader should not put too a high price on the edges
 - otherwise the follower will not buy them
- But the leader needs to put sufficiently high prices to optimize revenue



Minimum Spanning Tree problem

Minimum Spanning Tree (MST) problem

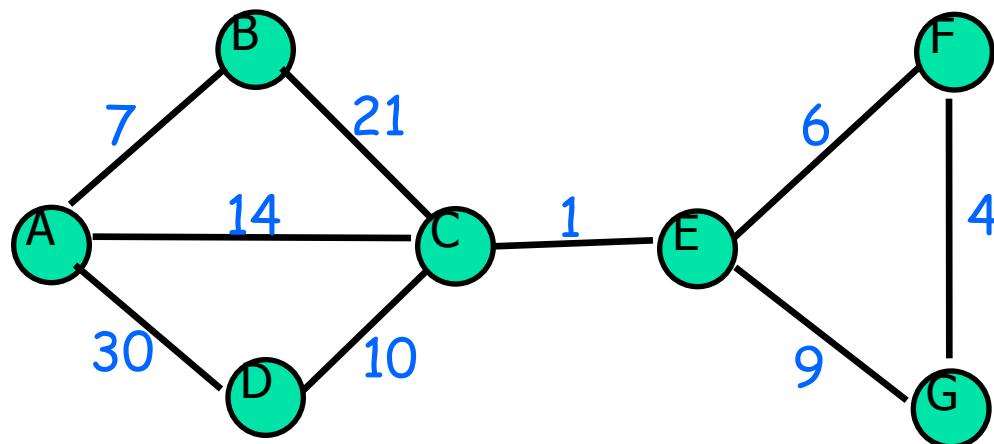
- **Input:**
 - undirected weighted graph $G=(V,E,w)$
- **Solution:**
 - a **spanning tree** of G , i.e. a tree $T=(V,F)$ with $F \subseteq E$
- **Measure** (to minimize):
 - Total weight of T : $\sum_{e \in F} w(e)$



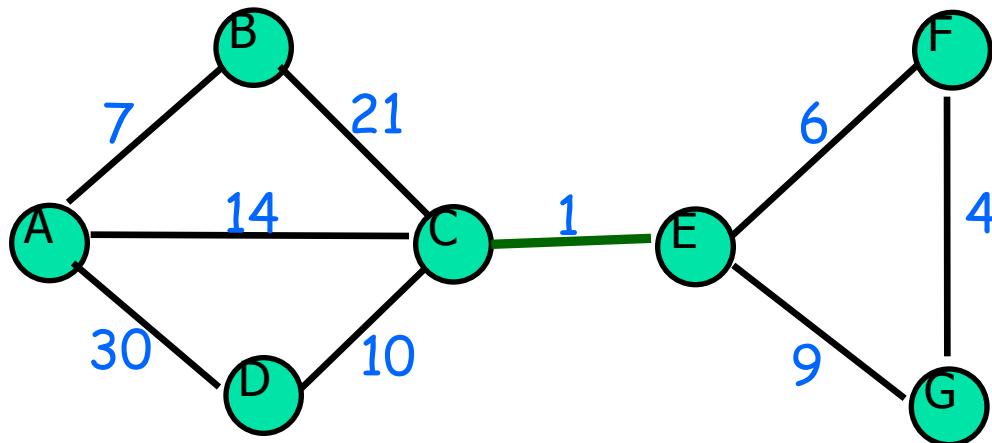
A famous algorithm: Kruskal's algorithm (1956)

- Start with an empty tree T
- consider the edges of G in non-decreasing order:
 - add the current edge e to T iff e does not form a cycle with the previous selected edges

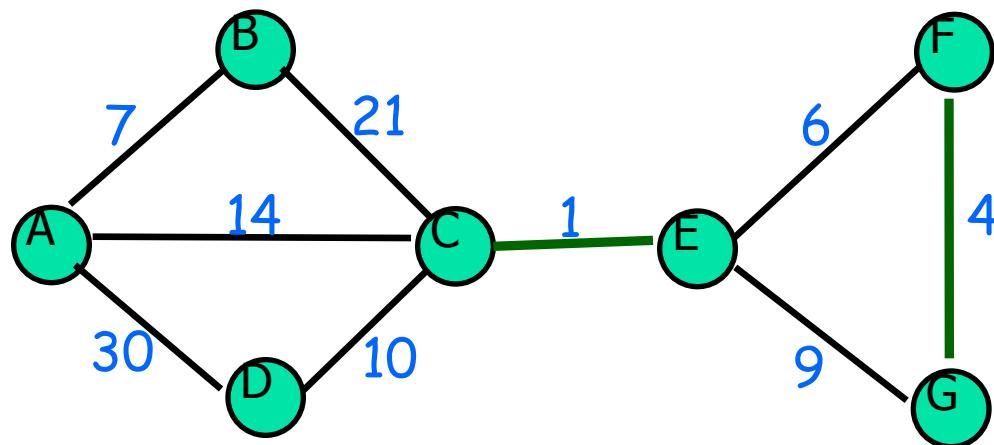
Example



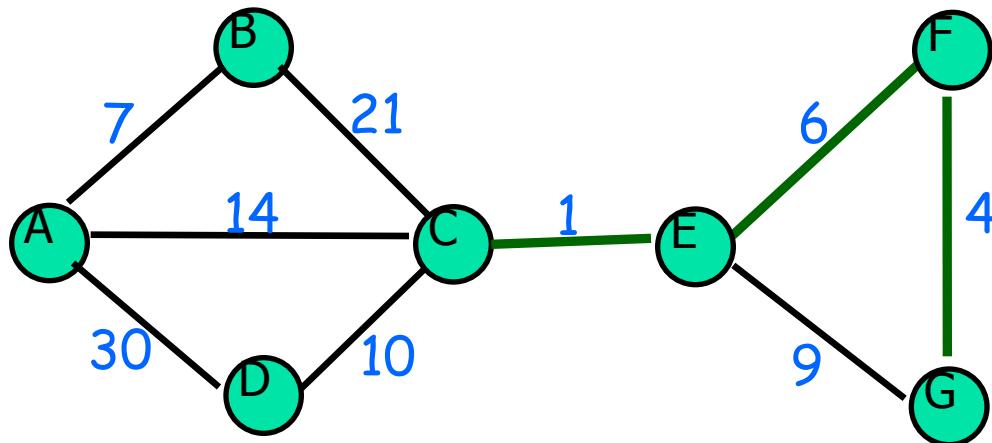
Example



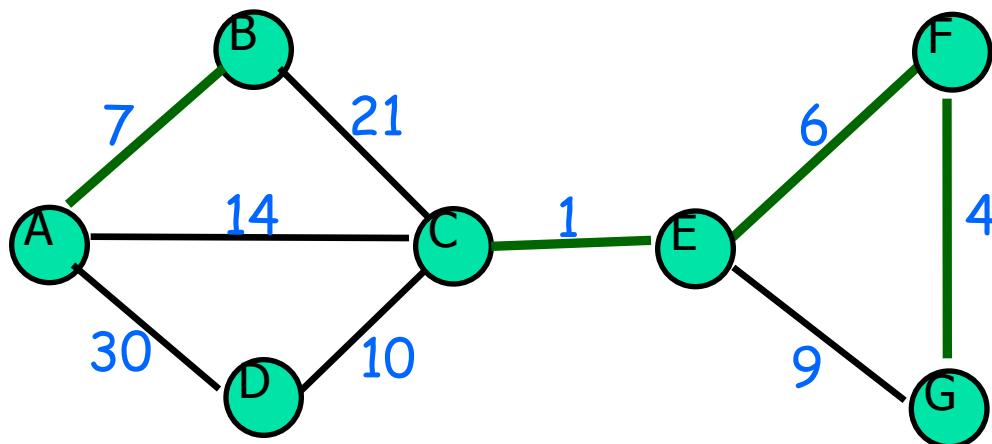
Example



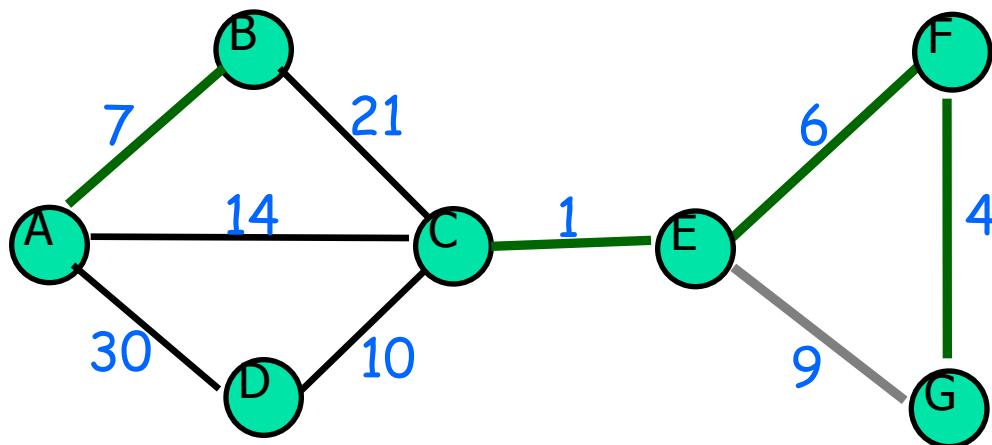
Example



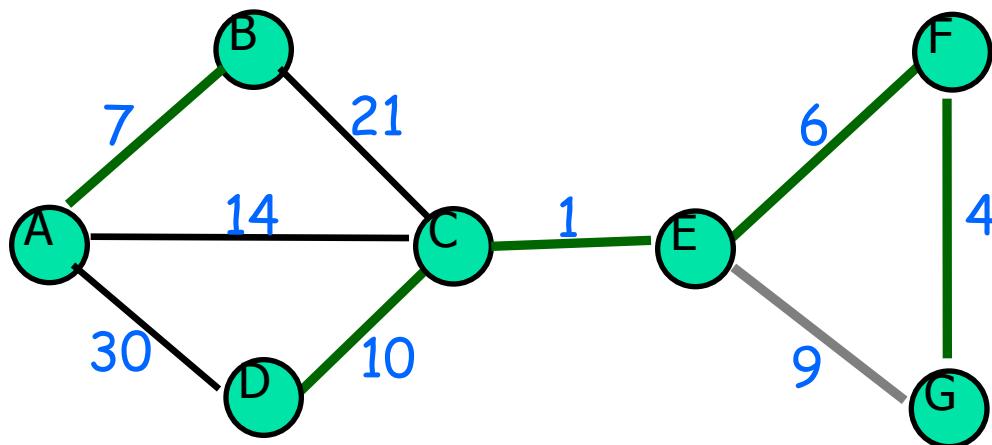
Example



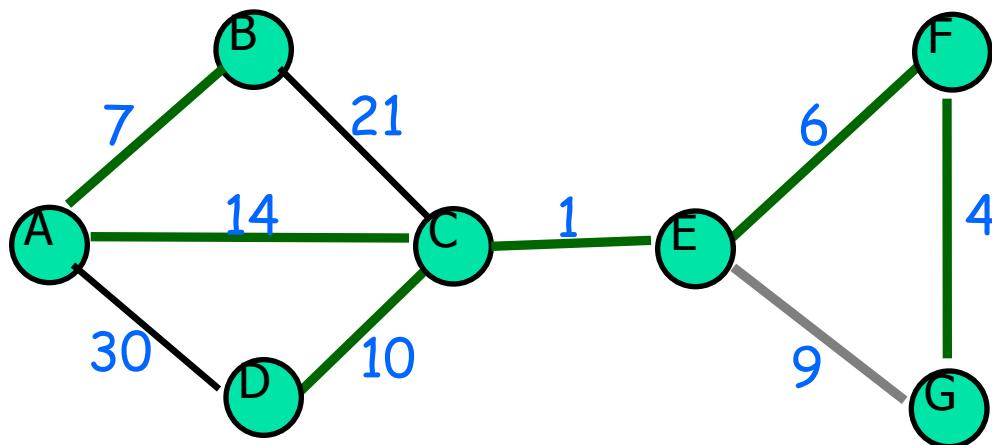
Example



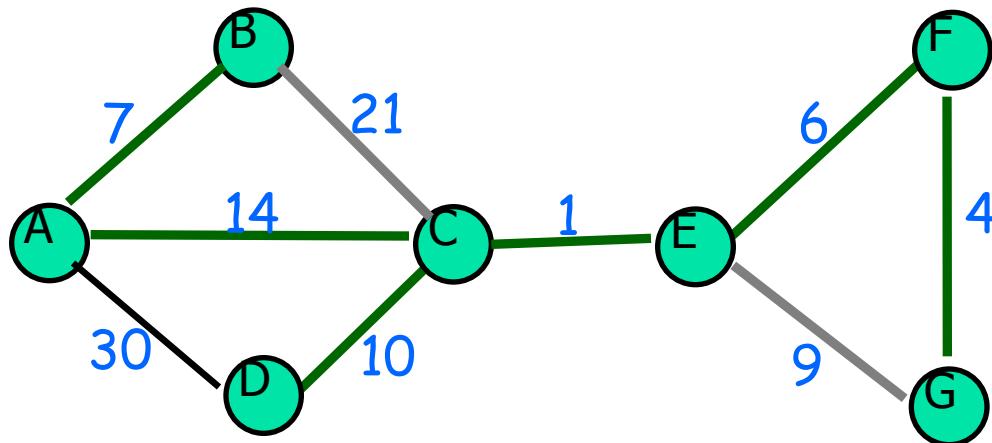
Example



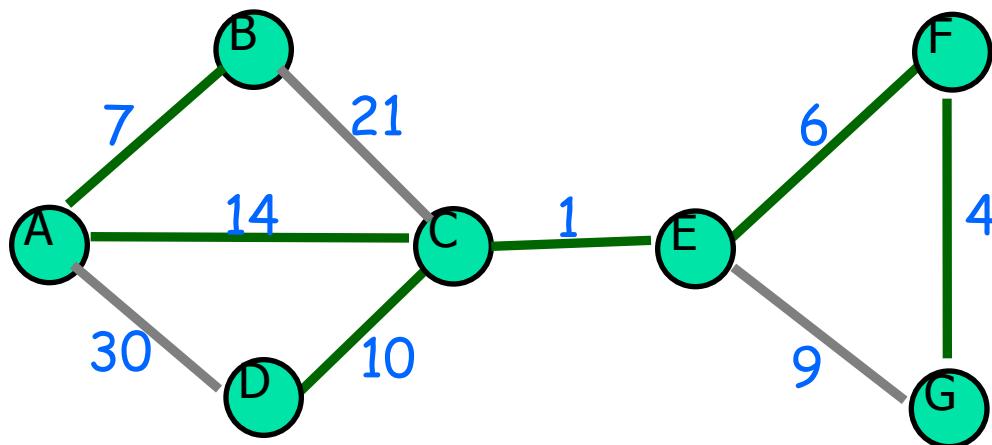
Example



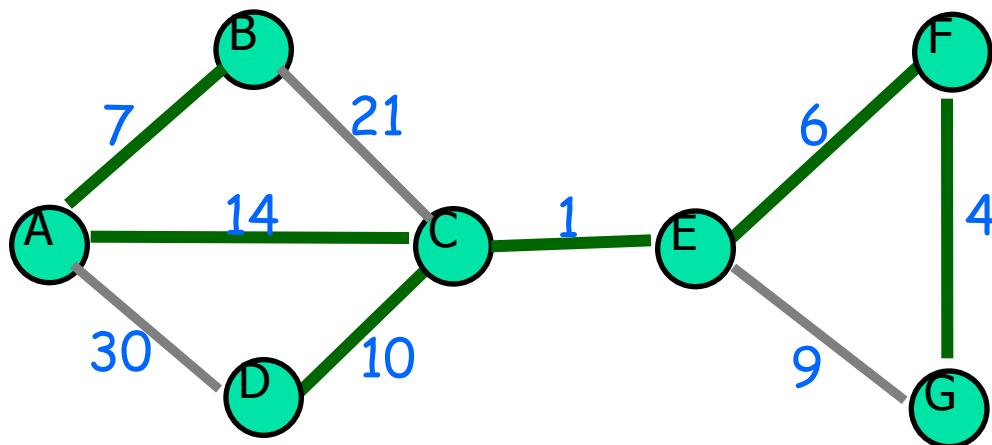
Example

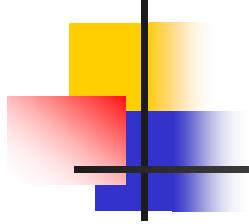


Example



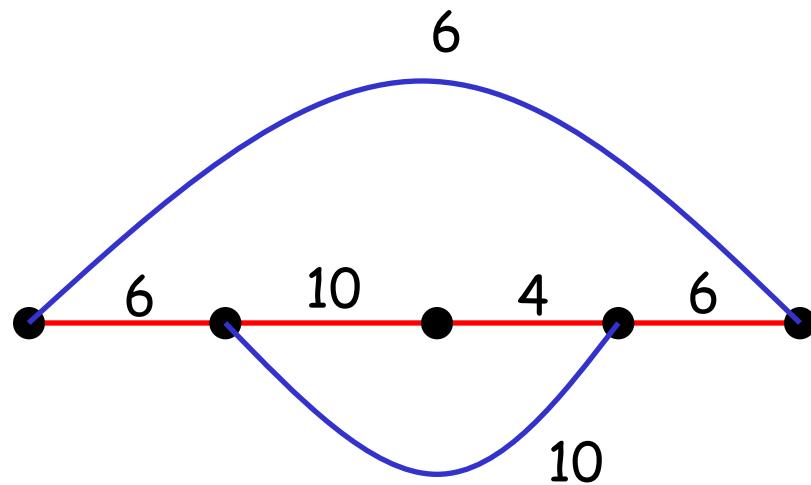
Example



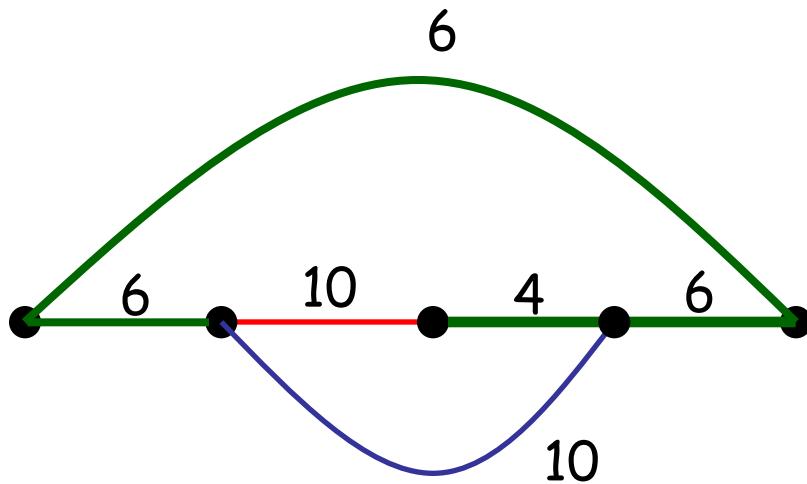


...turning to the Stackelberg
MST Game

Example

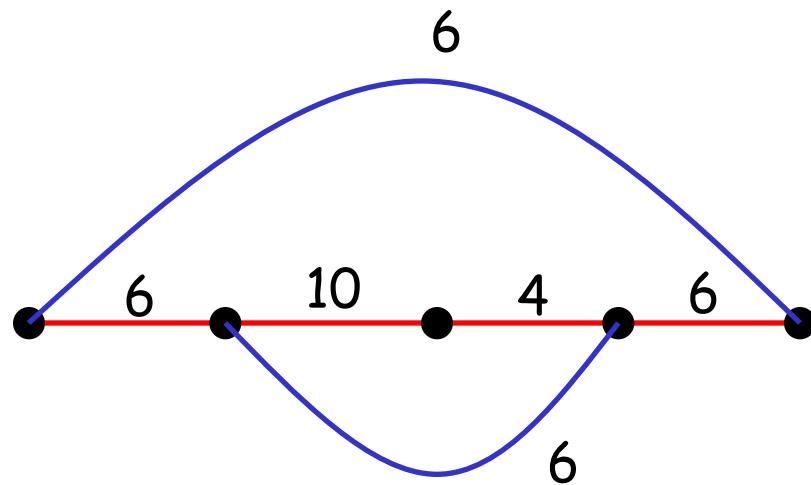


Example



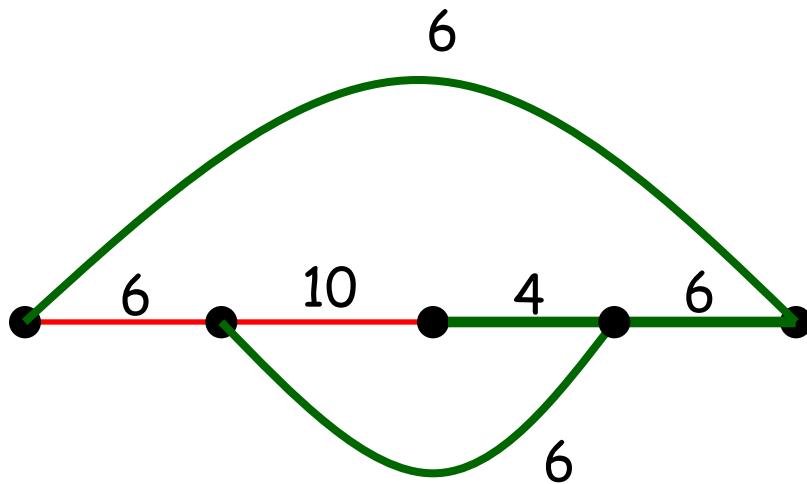
The revenue is 6

Example



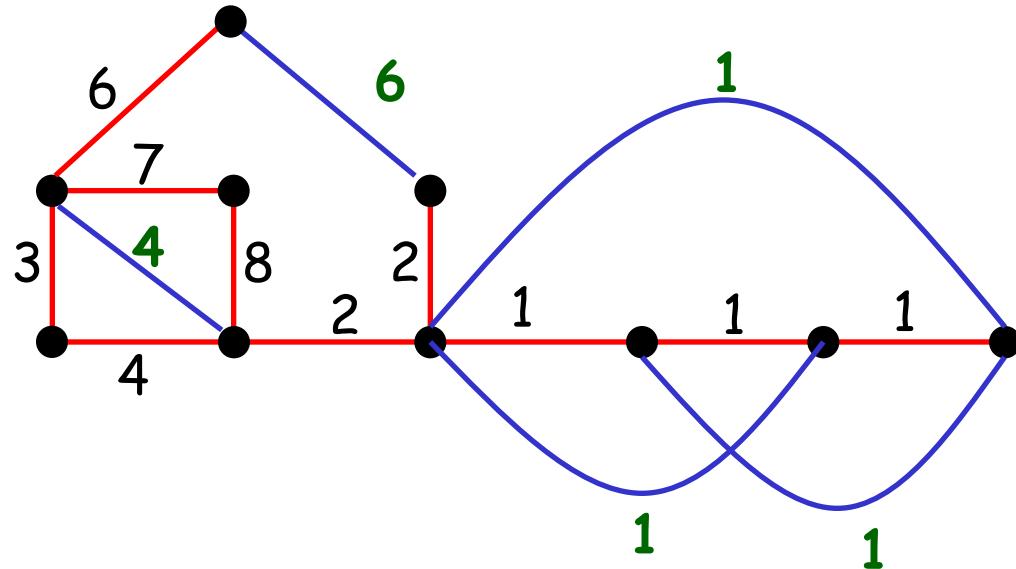
A better pricing...

Example

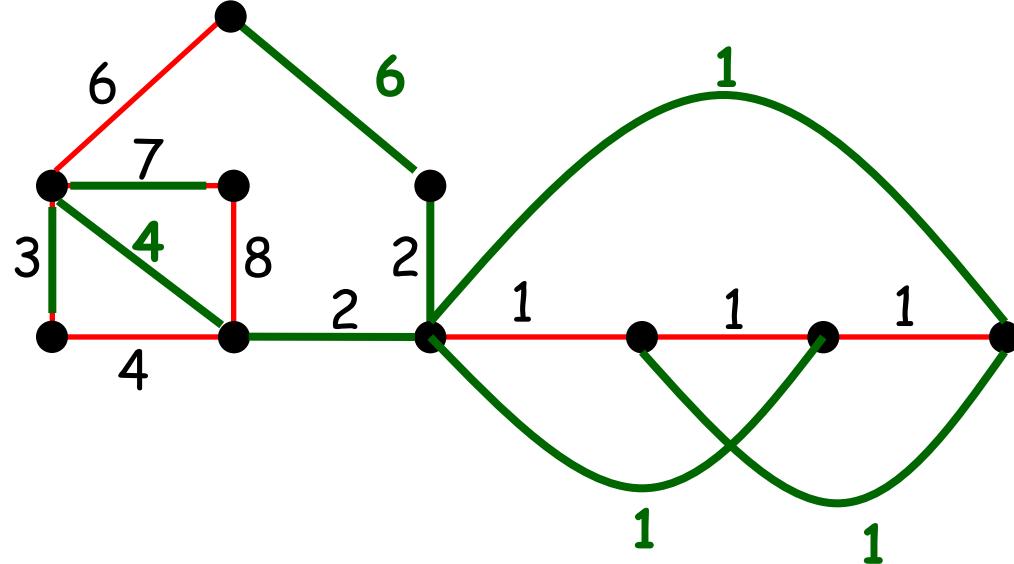


...with revenue 12

One more example

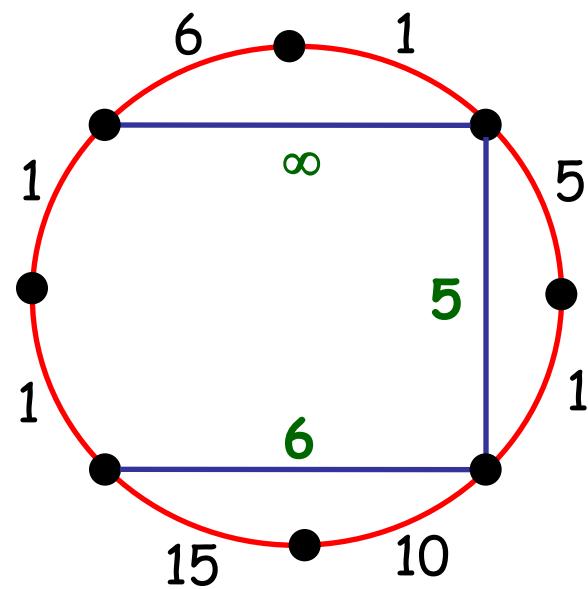


One more example

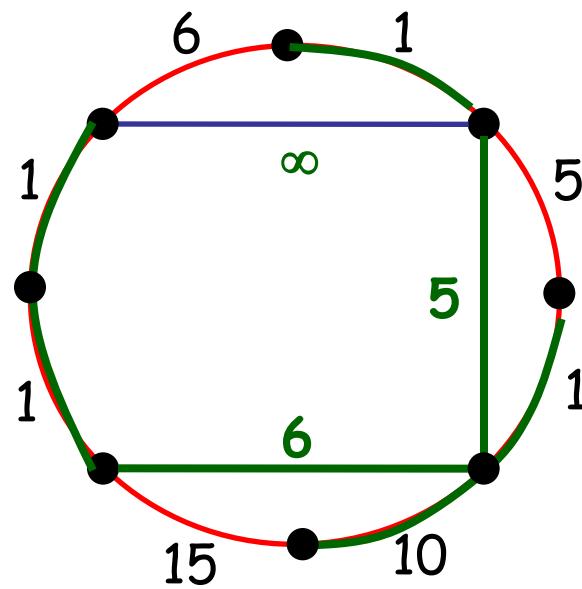


The revenue is 13

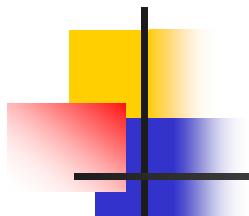
One more example



One more example



The revenue is 11



Assumptions

- G contains a spanning tree whose edges are all red
 - Otherwise the optimal revenue is unbounded
- Among all edges of the same weight, blue edges are always preferred to red edges
 - If we can get revenue r with this assumption, then we can get revenue $r-\varepsilon$, for any $\varepsilon > 0$
 - by decreasing prices suitably

Results:

Theorem

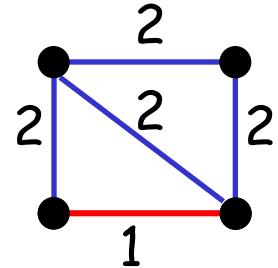
The Stackelberg MST game is NP-hard, even when $c(e) \in \{1, 2\}$ for all $e \in R$.

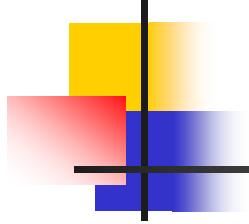
Theorem

There exists a polynomial-time ρ -approximated algorithm for StackelbergMST with $\rho = 1 + \min\{\log|B|, \log(n-1), \log(c_k/c_1)\}$, where c_1 and c_k are the minimum and maximum cost of a red edge.

The revenue of the leader depends on the **price function p** and not on the particular MST picked by the follower

- Let $w_1 < w_2 < \dots < w_h$ be the different edge weights
- The greedy(Kruskal's) algorithm works in h phases
- In its phase i , it considers:
 - all blue edges of weight w_i (if any)
 - Then, all red edges of weight w_i (if any)
- Number of selected blue edges of weight w_i does not depend on the order on which red and blue edges are considered!
- This implies...





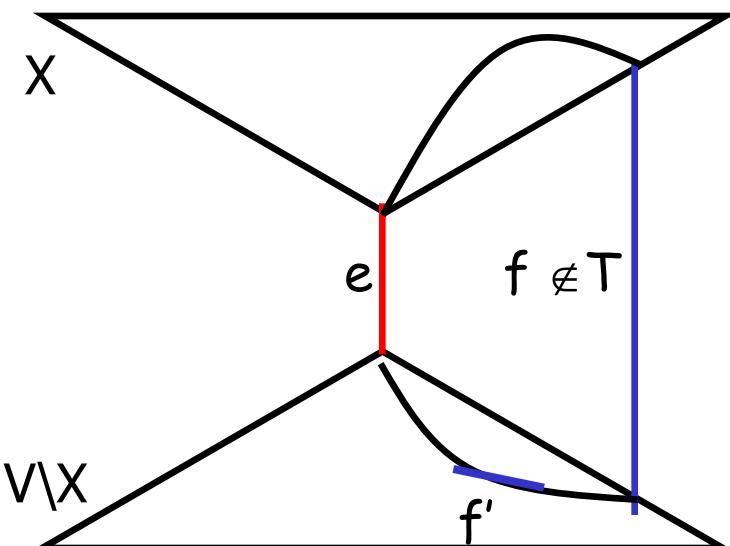
Lemma 1

In every optimal price function, the prices assigned to blue edges appearing in some MST belong to the set $\{c(e): e \in R\}$

Lemma 2

Let p be an optimal price function and T be the corresponding MST. Suppose that there exists a red edge e in T and a blue edge f not in T such that e belongs to the unique cycle C in $T+f$. Then there exists a blue edge f' distinct to f in C such that $c(e) < p(f') \leq p(f)$

proof



$$c(e) < p(f)$$

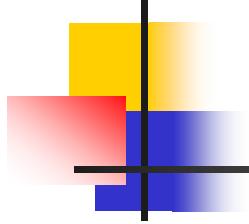
f' : the heaviest blue edge in C
(different to f)

$$p(f') \leq p(f)$$

if $p(f') \leq c(e)$...

... $p(f) = c(e)$ will imply a greater revenue

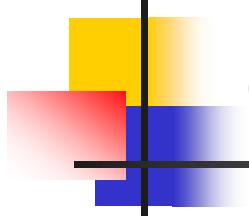




Theorem

The Stackelberg MST game is NP-hard, even when $c(e) \in \{1, 2\}$ for all $e \in R$

reduction from Set cover problem



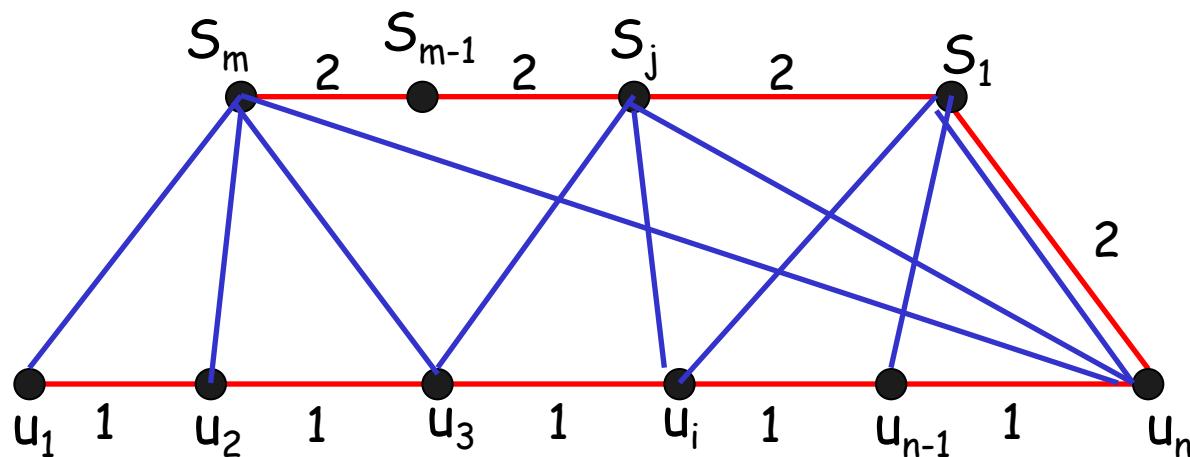
minimum Set Cover Problem

- INPUT:
 - Set of objects $U = \{u_1, \dots, u_n\}$
 - $\mathcal{S} = \{S_1, \dots, S_m\}$, $S_j \subseteq U$
- OUTPUT:
 - A cover $C \subseteq \mathcal{S}$ whose union is U and $|C|$ is minimized

$$U = \{u_1, \dots, u_n\} \quad S = \{S_1, \dots, S_m\}$$

w.l.o.g. we assume:
 $u_n \in S_j$, for every j

We define the following graph:

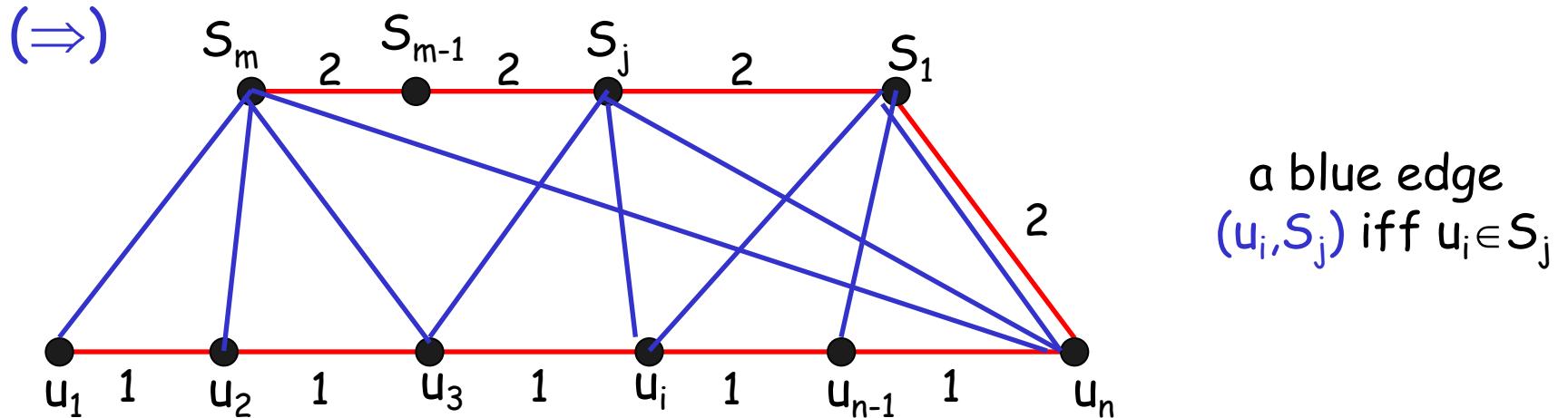


a blue edge
 (u_i, S_j) iff $u_i \in S_j$

Claim:

(U, S) has a cover of size at most $t \iff$

$$\text{maximum revenue } r^* \geq n+t-1+2(m-t) = n+2m-t-1$$



We define the price function as follows:

For every blue edge $e = (u_i, S_j)$,
 $p(e) = 1$ if S_j is in the cover, 2 otherwise

→ revenue $r = n + t - 1 + 2(m - t)$



(\Leftarrow)

p : optimal price function $p: B \rightarrow \{1, 2, \infty\}$ such that the corresponding MST T minimizes the number of red edges

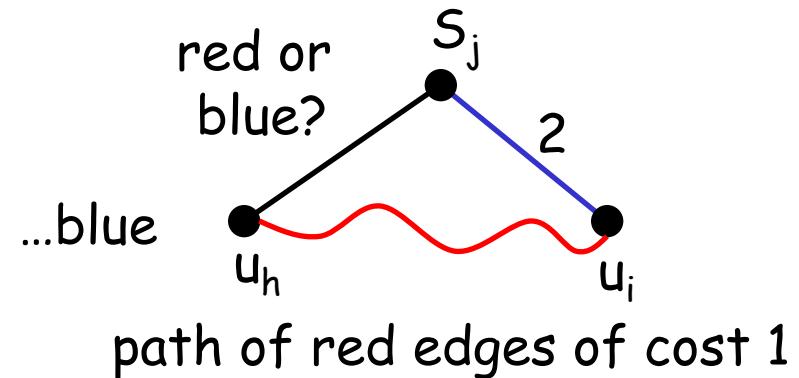
We'll show that:

1. T has blue edges only
2. There exists a cover of size at most t

Remark:

If all red edges in T have cost 1, then for every blue edge $e = (u_i, S_j)$ in T with price 2, we have that S_j is a leaf in T

by contradiction...



\rightarrow e cannot belong to T

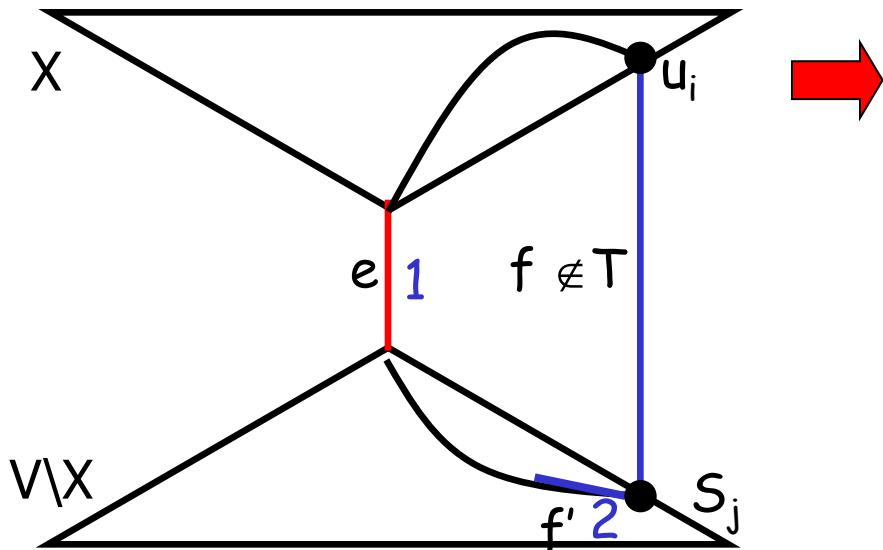
$(\Leftarrow), (1)$

e : heaviest red edge in T

since (V, B) is connected,
there exists blue edge $f \notin T$...

Lemma 2:

$\exists f' \neq f$ such that $c(e) < p(f') \leq p(f)$



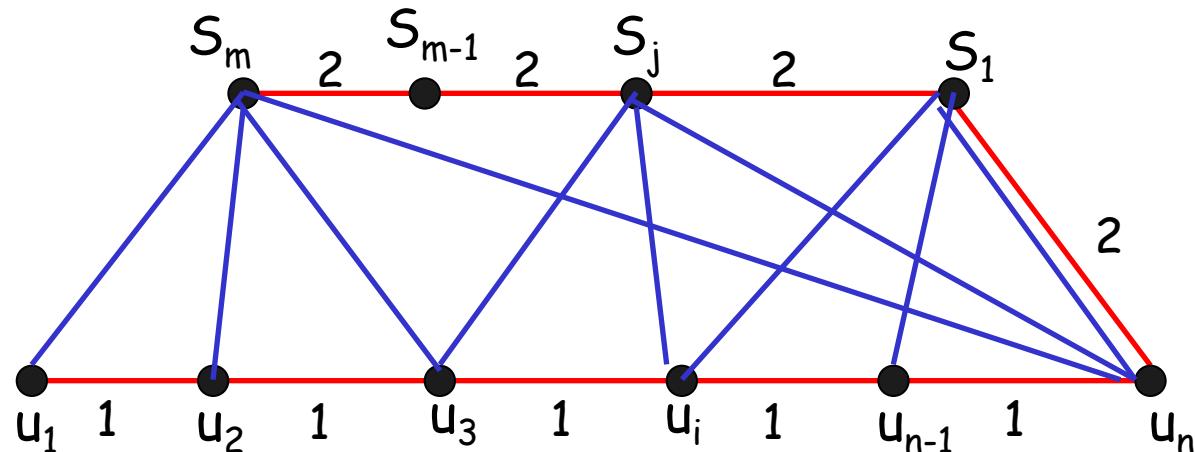
$c(e)=1$ and $p(f')=2$

By previous remark...
all blue edges in $C - \{f, f'\}$
have price 1

$\rightarrow p(f)=1$ and $p(f')=1$ leads to a new MST
with same revenue and less red edges.
A contradiction.



$(\Leftarrow), (2)$



Assume T contains no red edge

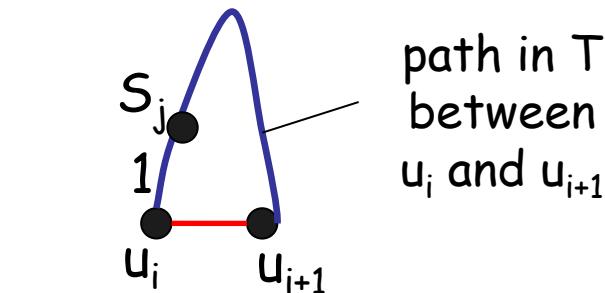
We define:

$C = \{S_j : S_j \text{ is linked to some blue edge in } T \text{ with price 1}\}$

every u_i must be incident in T to some blue edge of price 1

→ C is a cover

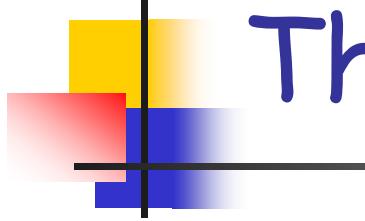
any $S_j \notin C$ must be a leaf in T



$$\text{revenue} = n + |C| - 1 + 2(m - |C|) = n + 2m - |C| - 1 \geq n + 2m - t - 1$$

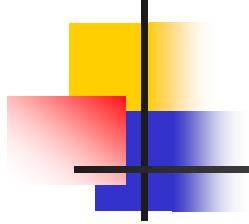
→ $|C| \leq t$





The single price algorithm

- Let $c_1 < c_2 < \dots < c_k$ be the different fixed costs
- For $i = 1, \dots, k$
 - set $p(e) = c_i$ for every $e \in B$
 - Look at the revenue obtained
- return the solution which gives the best revenue

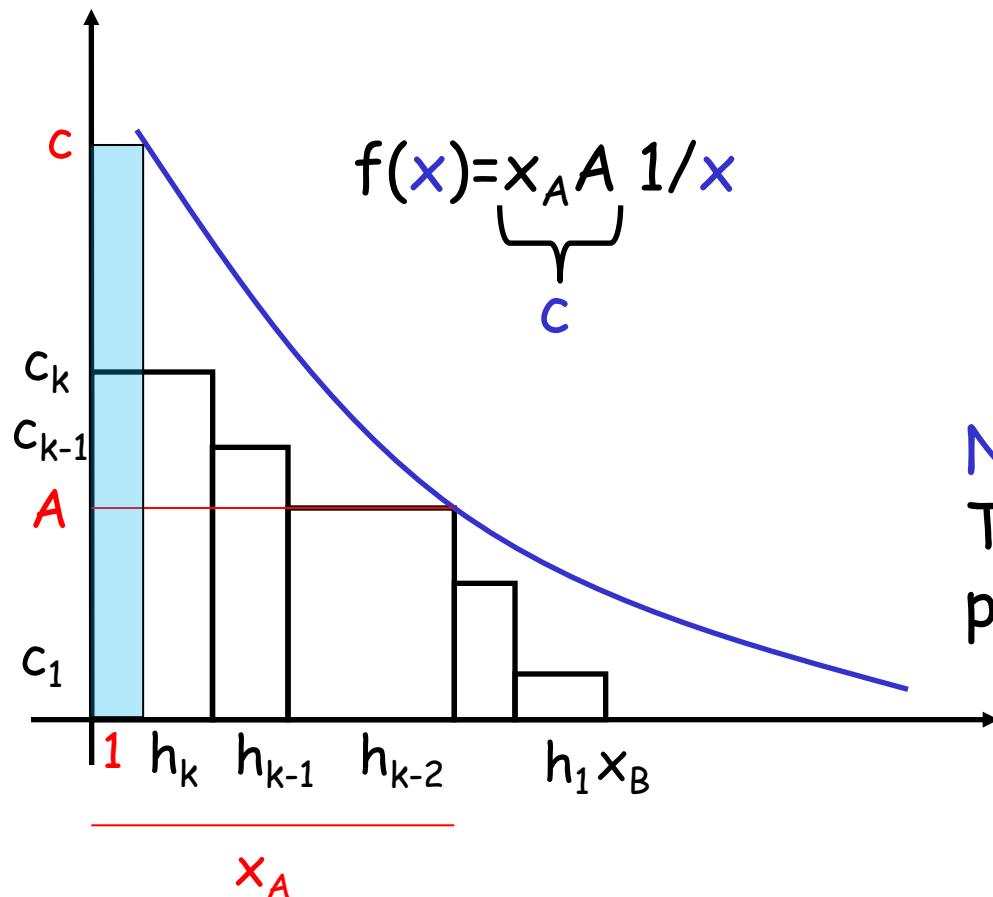


Theorem

Let r be the revenue of the single price algorithm; and let r^* be the optimal revenue. Then, $r^*/r \leq \rho$, where $\rho = 1 + \min\{\log|B|, \log(n-1), \log(c_k/c_1)\}$

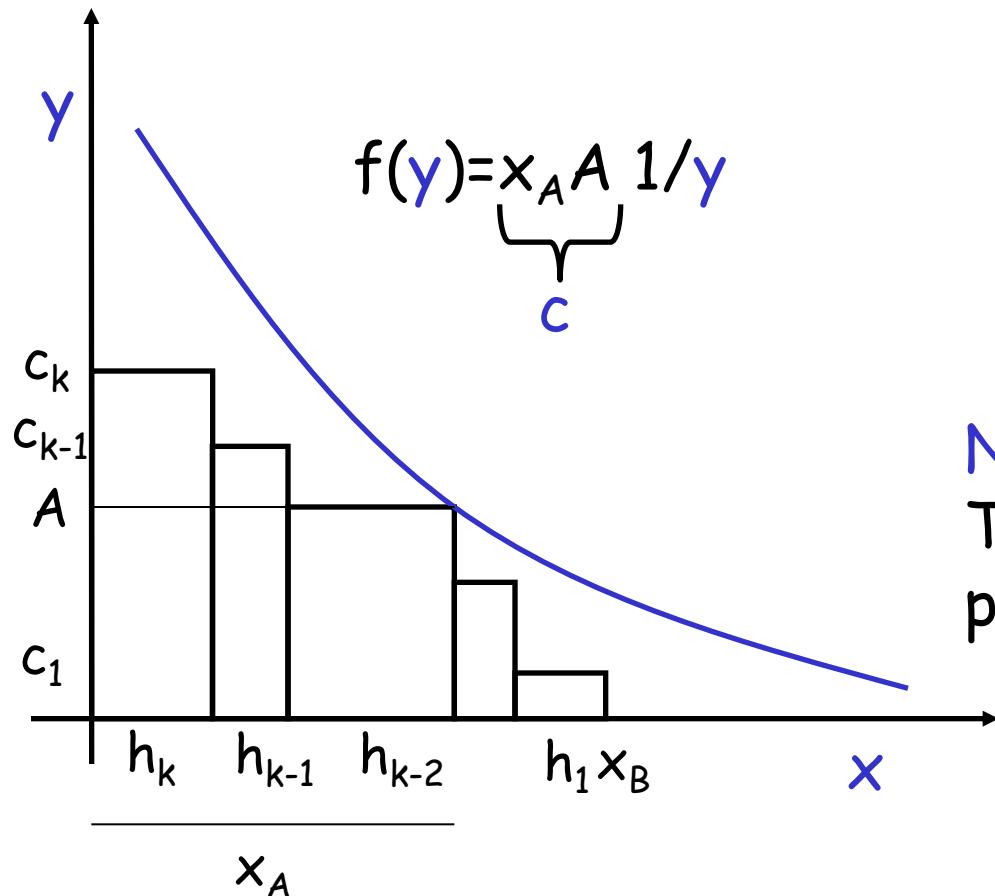
T : MST corresponding to the optimal price function

h_i : number of blue edges in T with price c_i



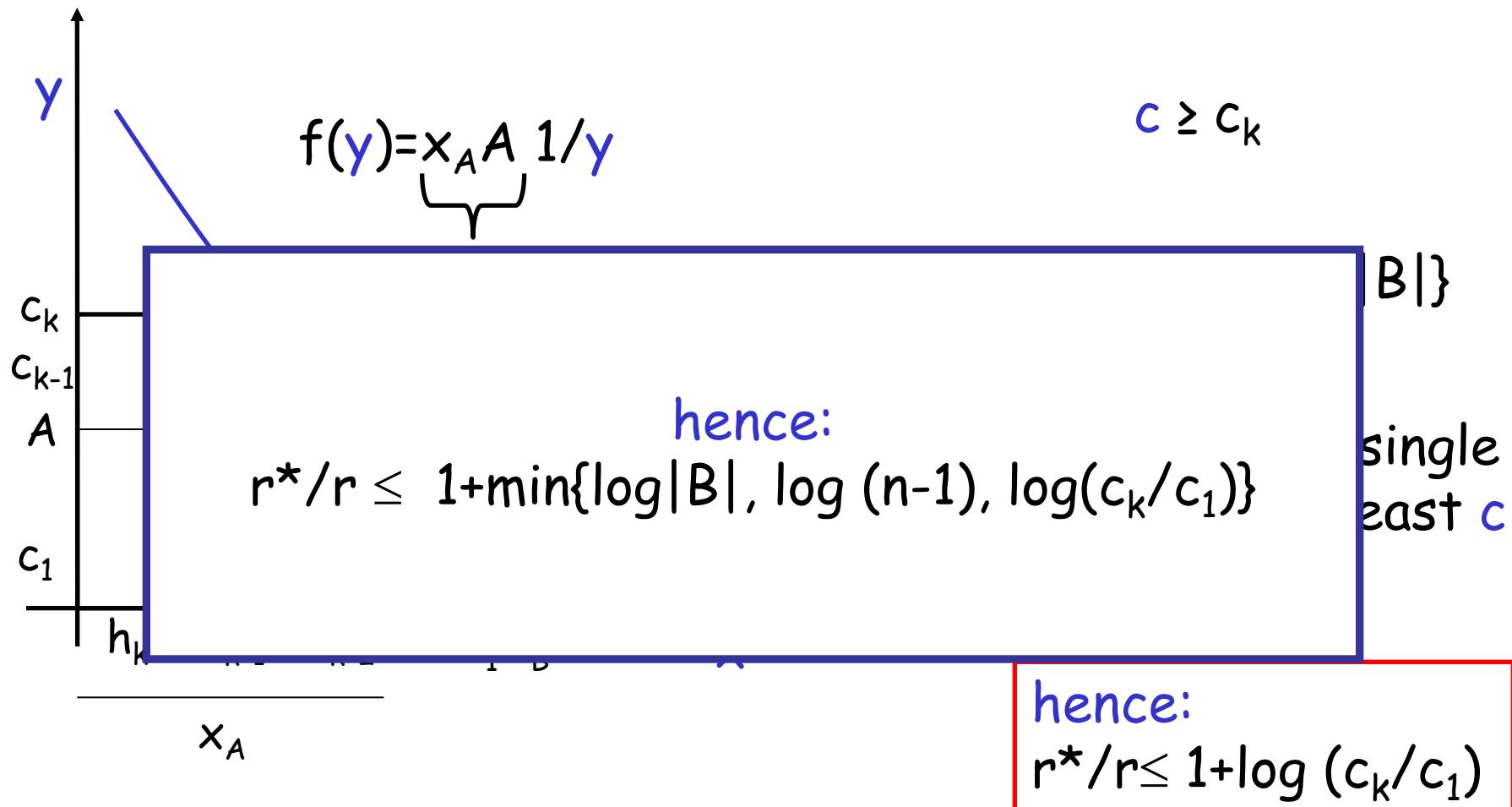
T : MST corresponding to the optimal price function

k_i : number of blue edges in T with price c_i



T : MST corresponding to the optimal price function

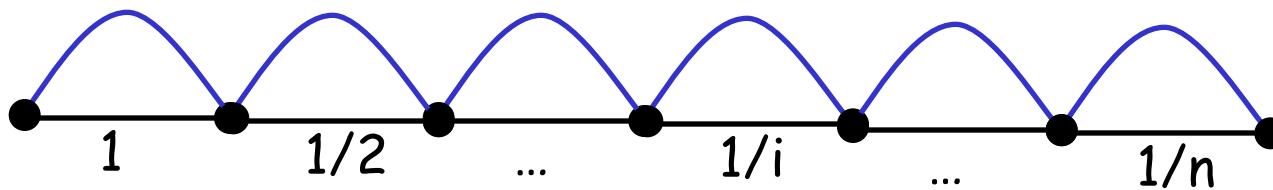
k_i : number of blue edges in T with price c_i



$$r^* \leq c + \int_{c_1}^{c_k} c \frac{1}{y} dy = c(1 + \log c_k - \log c_1) = c(1 + \log(c_k/c_1))$$



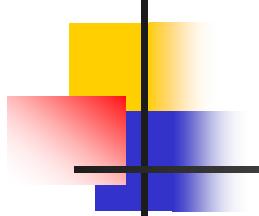
An asymptotically tight example



The single price algorithm obtains revenue $r=1$

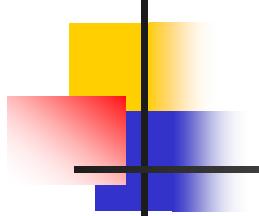
The optimal solution obtains revenue

$$r^* = \sum_{j=1}^n 1/j = H_n = \Theta(\log n)$$



Exercise: prove the following

Let r be the revenue of the single price algorithm; and let r^* be the optimal revenue. Then, $r^*/r \leq k$, where k is the number of distinct red costs



Exercise:

Give a polynomial time algorithm that, given an acyclic subset $F \subseteq B$, find a pricing p such that:

- (i) The corresponding MST T of p contains exactly F as set of blue edges, i.e. $E(T) \cap B = F$
- (ii) The revenue is maximized