Topic 4: Vertex Cover Problem

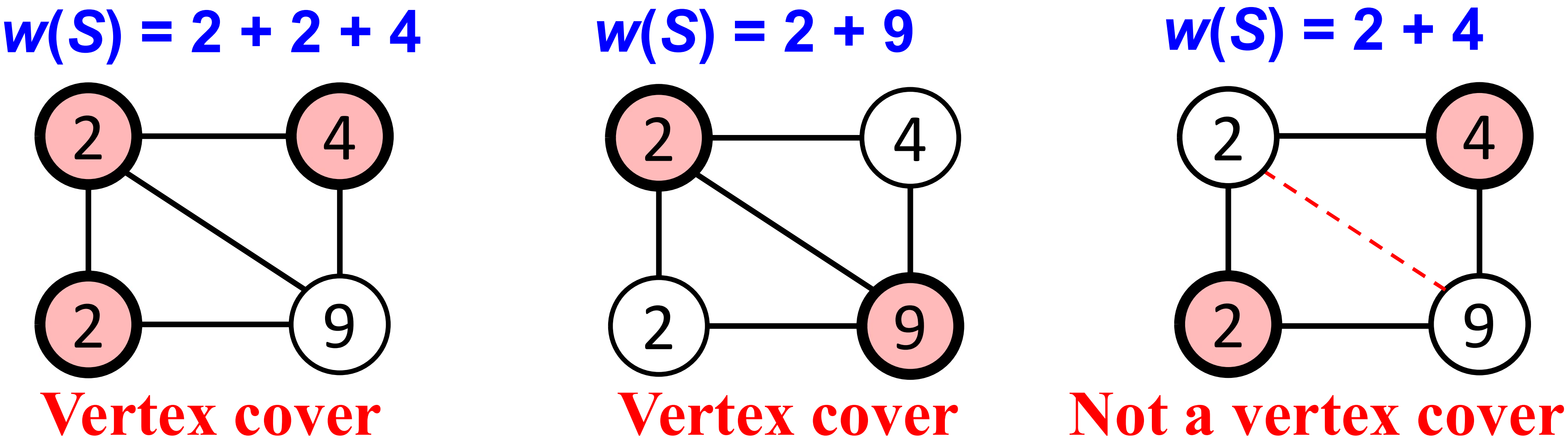
Input: Graph G: $G = (V, E)$

Weight of each vertex (node): w_i $(i \in V)$

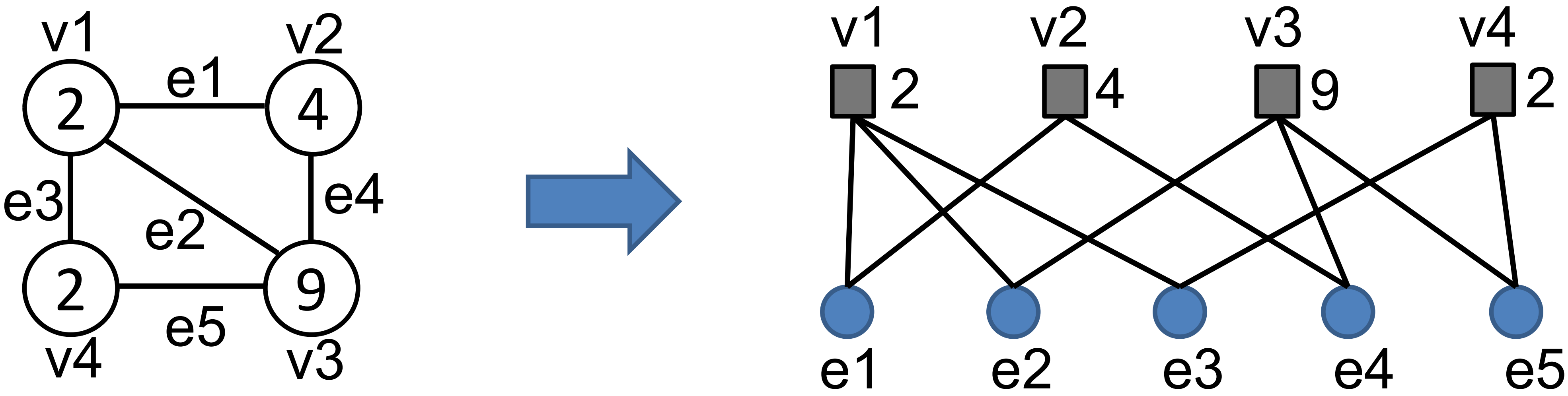
Output: Vertex cover S $(S \subset V)$ with the minimum total weight

Minimize $w(S) = \sum_{i \in S} w_i$

where S $(S \subset V)$ is a vertex cover (i.e., each edge in E has at least one end in S).

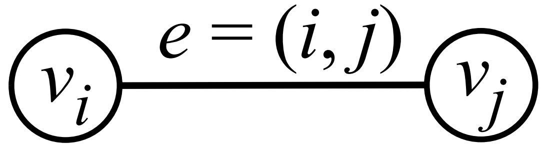


Vertex Cover Problem → Set Cover Problem



We can use the greedy set cover algorithm for the vertex cover problem, which is an $H(d^*)$-approximation algorithm where d^* is the maximum degree of the graph (i.e., the maximum number of edges from each vertex (node)).

Pricing Method (Idea)

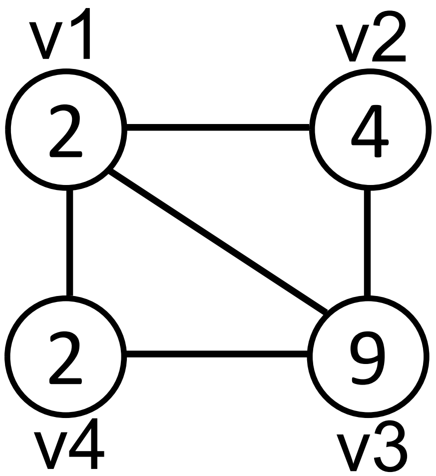


Edge e=(i, j) must be covered by vertex (node) v_i or v_j. Let p_e be the price that the edge e is willing to pay for being covered. The sum of prices over all edges incident (connected) to vertex v_i should be equal to or less than w_i (since they do not have to pay more than the total cost w_i and they can use other vertexes).

For each vertex v_i: $$\sum_{e=(i,j)} p_e \le w_i$$

$$p_{(i,j)} = p_{(j,i)}$$

If $\sum_{e=(i,j)} p_e = w_i$, **v_i is tight.**



$$v_1 : p_{(1,2)} + p_{(1,3)} + p_{(1,4)} \le w_1 = 2$$

$$v_2 : p_{(2,1)} + p_{(2,3)} \le w_2 = 4$$

$$v_3 : p_{(3,1)} + p_{(3,2)} + p_{(3,4)} \le w_3 = 9$$

$$v_4 : p_{(4,1)} + p_{(4,3)} \le w_4 = 2$$

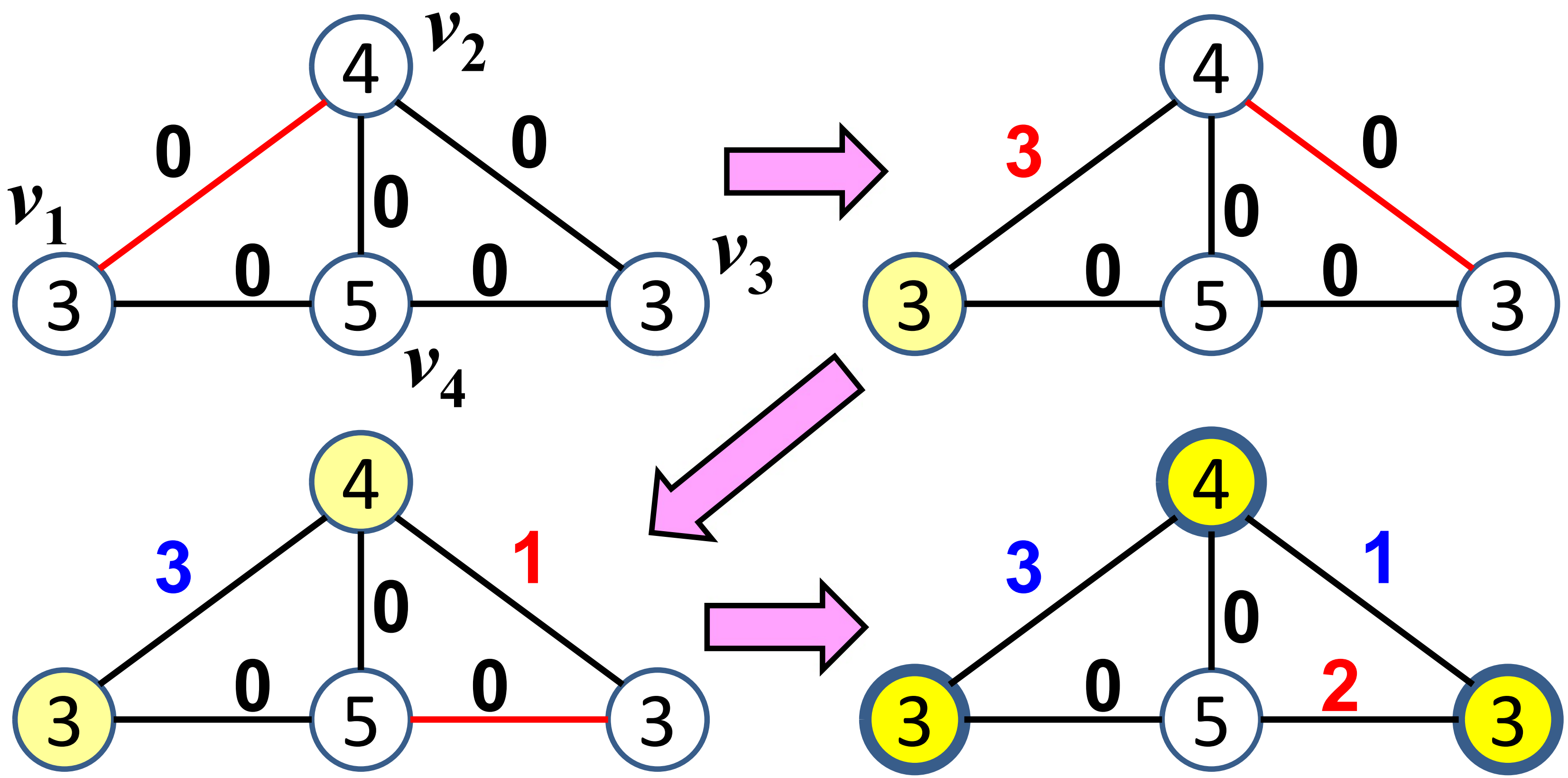
Pricing Method (Algorithm)

Initialization of p_e: $p_e = 0$ for each edge $e = (i, j)$.

Increase p_e: If neither vertex v_i nor v_j is tight, increase $p_{(i,j)}$ as much as possible under the condition: $\sum_{e=(i,j)} p_e \leq w_i$

(This condition is for both w_i and w_j)

Selection of a vertex cover S: Select all tight vertexes.



Pricing Method (Algorithm)

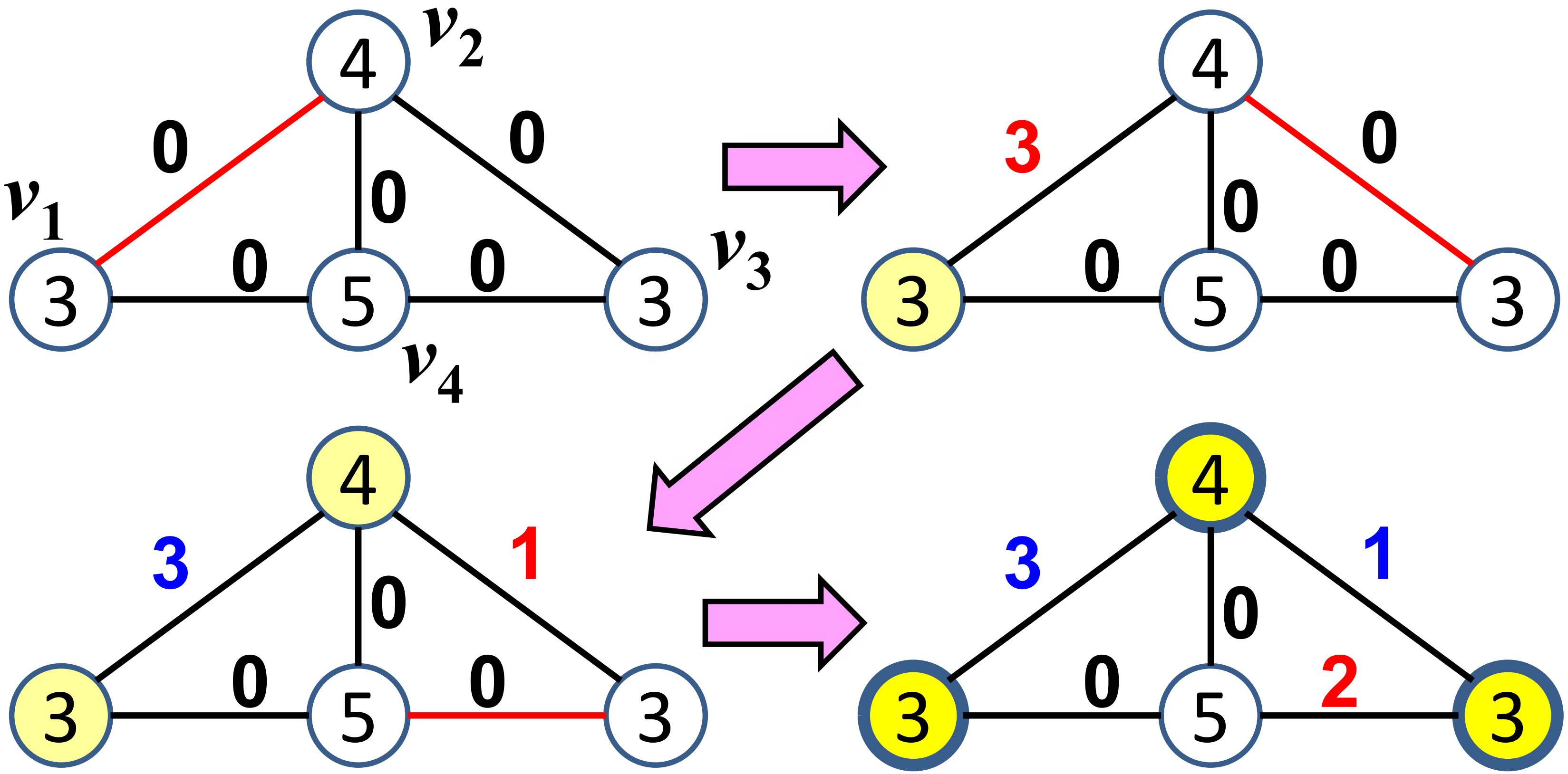
Which edge should be selected?

Initialization of p_e: $p_e = 0$ for each edge $e = (i, j)$.

Increase p_e: If neither vertex v_i nor v_j is tight, increase $p_{(i,j)}$ as much as possible under the condition: $\sum_{e=(i,j)} p_e \le w_i$

(This condition is for both w_i and w_j)

Selection of a vertex cover S: Select all tight vertexes.



Pricing Method (Algorithm)

We say a node i is *tight* (or "paid for") if $\sum_{e=(i,j)} p_e = w_i$.

procedure $\textsc{Vertex-Cover-Approx}(G, w)$
  Set $p_e = 0$ for all $e \in E$
  while $\exists$ edge $e = (i, j)$ such that neither i nor j is tight **do**
    Select e
    Increase p_e without violating fairness
  end while
  Let S = set of all tight nodes
  Return S.
end procedure

Fairness condition: $\sum_{e=(i,j)} p_e \le w_i$

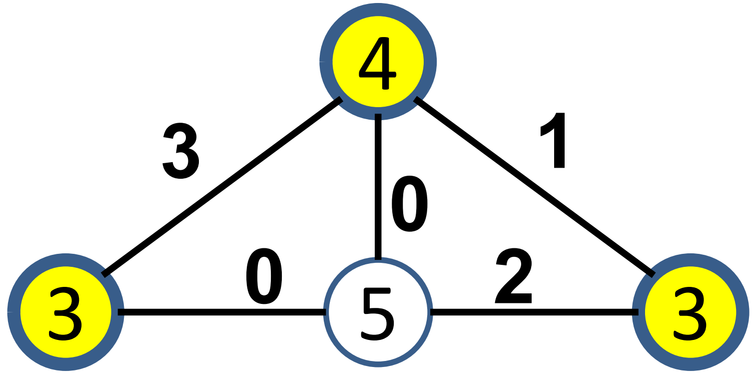
```
Weighted-Vertex-Cover-Approx(G, w) {
   foreach e in E
      pe = 0

   while (∃ edge i-j such that neither i nor j are tight)
      select such an edge e
      increase pe as much as possible until i or j tight
   }

   S ← set of all tight nodes
   return S
}
```

$$\sum_{e=(i,j)} p_e = w_i$$

Pricing Method (Analysis)



Example of the price assignment

Let us consider a price assignment satisfying the following condition:

Price Assignment: $\sum_{e=(i,j)} p_e \leq w_i$

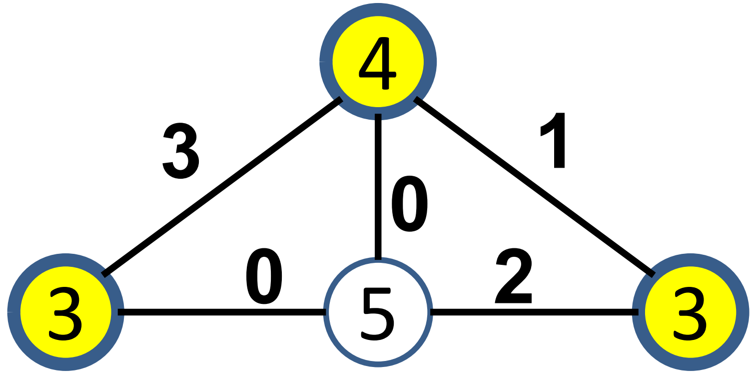
For the given price assignment, the following inequality relation holds for an arbitrary given vertex cover S.

For any vertex cover *S*: $\sum_{e \in E} p_e \leq \sum_{i \in S} \sum_{e=(i,j)} p_e \leq \sum_{i \in S} w_i = w(S)$

This is from the assumption

Since S is a vertex cover, all edges are covered (some edges can be counted twice).

Pricing Method (Analysis)



Price Assignment: $\sum_{e=(i,j)} p_e \le w_i$

For any vertex cover S: $\sum_{e \in E} p_e \le \sum_{i \in S} \sum_{e=(i,j)} p_e \le \sum_{i \in S} w_i = w(S)$

For the vertex cover S by the algorithm: $w(S) \le 2w(S^*)$

This is to be shown.

Since all vertexes v_i in S are tight,

$$\sum_{e=(i,j)} p_e = w_i \Rightarrow w(S) = \sum_{i \in S} w_i = \sum_{i \in S} \sum_{e=(i,j)} p_e \le 2 \sum_{e \in E} p_e$$

(An edge $e = (i, j)$ can be counted at most twice.)

Since $\sum_{e \in E} p_e \le w(S)$ holds for any vertex cover S including S^*,

$$w(S) \le 2 \sum_{e \in E} p_e \le 2w(S^*)$$

Homework

Exercise 9-1:

Create an example of the vertex cover problem where $w(S)$ obtained by the pricing method is always $2w(S^*)$ independent of the order of edges. Your example should include at least three vertexes.

Exercise 9-2:

Create an example of the vertex cover problem where better results are always obtained (independent of the order of edges and a tie-breaking mechanism) by the greedy set cover algorithm than the pricing method. Your example should include at least three vertexes.

Exercise 9-3:

Create an example of the vertex cover problem where better results are always obtained (independent of the order of edges and a tie-breaking mechanism) by the pricing method than the greedy set cover algorithm. Your example should include at least three vertexes.

Exercise 9-1:

Create an example of the vertex cover problem where $w(S)$ obtained by the pricing method is always $2w(S^*)$ independent of the order of edges. Your example should include at least three vertexes.

Please try to create interesting examples

Exercise 9-2:

Create an example of the vertex cover problem where better results are always obtained (independent of the order of edges and a tie-breaking mechanism) by the greedy set cover algorithm than the pricing method. Your example should include at least three vertexes.

Please try to create interesting examples

Exercise 9-3:

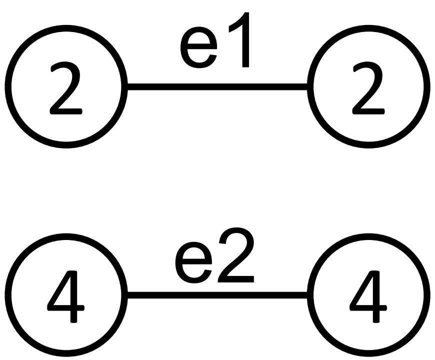
Create an example of the vertex cover problem where better results are always obtained (independent of the order of edges and a tie-breaking mechanism) by the pricing method than the greedy set cover algorithm. Your example should include at least three vertexes.

Not interesting example for Exercise 9.1 and Exercise 9.2.

9.1: The pricing algorithm result $w(S)$ is always $2w(S^*)$ independent of the order of edges.

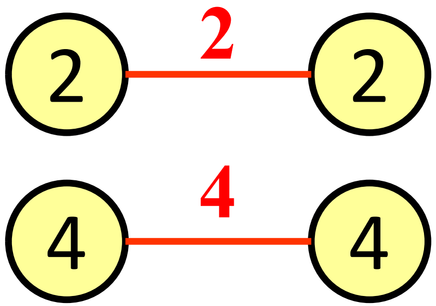
9.2: The greedy set cover result is always better than the pricing algorithm result independent of the order of edges and a tie-breaking mechanism.

Example



Optimal value: $w(S^*) = 6$

Pricing algorithm result: $w(S) = 12$



Greedy set cover result: $w(S) = 6$