Set Cover
CSCI 532
Handling NP-Completeness
Approximation Algorithms

\[ \text{ALG} \leq \alpha \text{OPT} \]

Cost (size) of algorithm’s solution.  
Approximation Ratio  
Cost (size) of optimal solution.
Vertex Cover

**VC 2-approximation algorithm:**

while uncovered edge exists
    select both vertices from uncovered edge

An edge is uncovered if it does not share vertices with any previously selected edges. Let $E'$ be the edges selected by the algorithm.

⇒ # vertices selected by algorithm = ALG = 2 $|E'|$

A vertex from each edge in $E'$ must be part of every vertex cover.

⇒ $|E'| \leq \text{OPT}$

Therefore, $\text{ALG} = 2 |E'| \leq 2 \text{OPT} \implies \text{ALG} \leq 2 \text{OPT}$
Set Cover

Set Cover: Given a set of elements (the universe), and sets containing those elements, find the smallest number of sets so that every element of the universe is included.

Example:
Set Cover

Set Cover: Given a set of elements (the universe), and sets containing those elements, find the smallest number of sets so that every element of the universe is included.

Example:

\[ U = \{1, 4, 7, 8, 10\} \]
\[ S = \{\{1, 7, 8\}, \{1, 4, 7\}, \{7, 8\}, \{4, 8, 10\}\} \]
Set Cover

Set Cover: Given a set of elements (the universe), and sets containing those elements, find the smallest number of sets so that every element of the universe is included.

Example:

\[ U = \{1, 4, 7, 8, 10\} \]
\[ S = \{\{1, 7, 8\}, \{1, 4, 7\}, \{7, 8\}, \{4, 8, 10\}\} \]

\[ \{\{1, 7, 8\}, \{4, 8, 10\}\} \]  \[ \{\{1, 4, 7\}, \{7, 8\}\} \]
Set Cover

Set Cover: Given a set of elements (the universe), and sets containing those elements, find the smallest number of sets so that every element of the universe is included.

Example:
Set Cover

Set Cover: Given a set of elements (the universe), and sets containing those elements, find the smallest number of sets so that every element of the universe is included.

Algorithm:
Set Cover

Set Cover: Given a set of elements (the universe), and sets containing those elements, find the smallest number of sets so that every element of the universe is included.

Greedy Algorithm:

while element of universe not included
select $S_i$ with largest number of excluded elements.
Set Cover

Set Cover: Given a set of elements (the universe), and sets containing those elements, find the smallest number of sets so that every element of the universe is included.

Greedy Algorithm:

\[
\text{while element of universe not included} \\
\quad \text{select } S_i \text{ with largest number of excluded elements.}
\]

1. Valid?
2. Polynomial Time?
3. Performance?
Set Cover

Set Cover: Given a set of elements (the universe), and sets containing those elements, find the smallest number of sets so that every element of the universe is included.

Greedy Algorithm:

while element of universe not included
    select $S_i$ with largest number of excluded elements.

1. Valid. Every element of universe will be included.
2. Polynomial Time. $O(|S|^2|U|)$.
3. Performance?
Set Cover – Performance

Suppose the universe contains $n$ elements.
Set Cover – Performance

Suppose the universe contains $n$ elements.

$\text{ALG} \leq \alpha \text{OPT}$
Set Cover – Performance

Suppose the universe contains $n$ elements.

ALG $\leq \alpha$ OPT
Set Cover – Performance

Suppose the universe contains $n$ elements.
The first set selected will have $\geq \frac{n}{\text{OPT}}$ elements because?

ALG = # sets selected by the algorithm to cover all $n$ elements.
OPT = # sets in an optimal solution to cover all $n$ elements.
Suppose the universe contains $n$ elements.
The first set selected will have $\geq \frac{n}{\text{OPT}}$ elements because?

Claim: The first set selected will be the largest set.

Justification: ?
Set Cover – Performance

Suppose the universe contains $n$ elements. The first set selected will have $\geq \frac{n}{\text{OPT}}$ elements because?

Claim: The first set selected will be the largest set.

Justification: At each iteration, we cover the largest number of uncovered elements, and all the elements are uncovered in the first iteration.
Set Cover – Performance

Suppose the universe contains $n$ elements. The first set selected will have $\geq \frac{n}{\text{OPT}}$ elements because?

Claim: The first set selected will be the largest set.

Justification: At each iteration, we cover the largest number of uncovered elements, and all the elements are uncovered in the first iteration.

Suppose every set has $< \frac{n}{\text{OPT}}$ elements.
Set Cover – Performance

Suppose the universe contains $n$ elements.
The first set selected will have $\geq \frac{n}{\text{OPT}}$ elements because?

Claim: The first set selected will be the largest set.

Justification: At each iteration, we cover the largest number of uncovered elements, and all the elements are uncovered in the first iteration.

Suppose every set has $< \frac{n}{\text{OPT}}$ elements. An optimal solution exists that uses OPT sets,
Set Cover – Performance

Suppose the universe contains $n$ elements. The first set selected will have $\geq \frac{n}{\text{OPT}}$ elements because?

Claim: The first set selected will be the largest set.

Justification: At each iteration, we cover the largest number of uncovered elements, and all the elements are uncovered in the first iteration.

Suppose every set has $< \frac{n}{\text{OPT}}$ elements. An optimal solution exists that uses OPT sets, so that optimal solution covers $< \frac{n}{\text{OPT}}$ OPT = $n$ elements.
Set Cover – Performance

Suppose the universe contains $n$ elements. The first set selected will have $\geq \frac{n}{\text{OPT}}$ elements because?

Claim: The first set selected will be the largest set.

Justification: At each iteration, we cover the largest number of uncovered elements, and all the elements are uncovered in the first iteration.

Suppose every set has $< \frac{n}{\text{OPT}}$ elements. An optimal solution exists that uses OPT sets, so that optimal solution covers $< \frac{n}{\text{OPT}}$ OPT = $n$ elements. But that means it is not a valid solution.
Set Cover – Performance

Suppose the universe contains $n$ elements.
The first set selected will have $\geq \frac{n}{\text{OPT}}$ elements because the first set selected is the largest one.

Claim: The first set selected will be the largest set.

Justification: At each iteration, we cover the largest number of uncovered elements, and all the elements are uncovered in the first iteration.

Suppose every set has $< \frac{n}{\text{OPT}}$ elements. An optimal solution exists that uses OPT sets, so that optimal solution covers $< \frac{n}{\text{OPT}} \text{OPT} = n$ elements. But that means it is not a valid solution.
Set Cover – Performance

Suppose the universe contains \( n \) elements.
The first set selected will have \( \geq \frac{n}{\text{OPT}} \) elements because the first set selected is the largest one and if all sets had fewer than \( \frac{n}{\text{OPT}} \) elements, there would be no way to cover all \( n \) elements with only \( \text{OPT} \) sets.

Claim: The first set selected will be the largest set.

Justification: At each iteration, we cover the largest number of uncovered elements, and all the elements are uncovered in the first iteration.

Suppose every set has \( < \frac{n}{\text{OPT}} \) elements. An optimal solution exists that uses \( \text{OPT} \) sets, so that optimal solution covers \( < \frac{n}{\text{OPT}} \) \( \text{OPT} = n \) elements. But that means it is not a valid solution.
Set Cover – Performance

Suppose the universe contains $n$ elements.
The first set selected will have $\geq \frac{n}{\text{OPT}}$ elements.
Then, the number of elements remaining after the first iteration is: 

? 

ALG = # sets selected by the algorithm to cover all $n$ elements.
OPT = # sets in an optimal solution to cover all $n$ elements.
Set Cover – Performance

Suppose the universe contains \( n \) elements. The first set selected will have \( \geq \frac{n}{\text{OPT}} \) elements. Then, the number of elements remaining after the first iteration is:

\[
    n_1 \leq n - \frac{n}{\text{OPT}} = n \left( 1 - \frac{1}{\text{OPT}} \right)
\]

ALG = # sets selected by the algorithm to cover all \( n \) elements.
OPT = # sets in an optimal solution to cover all \( n \) elements.
Set Cover – Performance

Suppose the universe contains $n$ elements.
The first set selected will have $\geq \frac{n}{\text{OPT}}$ elements.
Then, the number of elements remaining after the first iteration is:

$$n_1 \leq n - \frac{n}{\text{OPT}} = n \left(1 - \frac{1}{\text{OPT}}\right)$$

Some remaining set has at least $\frac{n_1}{\text{OPT}}$ still not covered elements because?
Set Cover – Performance

Suppose the universe contains $n$ elements. The first set selected will have $\geq \frac{n}{\text{OPT}}$ elements. Then, the number of elements remaining after the first iteration is:

$$n_1 \leq n - \frac{n}{\text{OPT}} = n \left(1 - \frac{1}{\text{OPT}}\right)$$

Some remaining set has at least $\frac{n_1}{\text{OPT}}$ still not covered elements because?

Case 1: The first set we picked is not in the optimal solution.

Case 2: The first set we picked is in the optimal solution.
Set Cover – Performance

Suppose the universe contains \( n \) elements. The first set selected will have \( \geq \frac{n}{\text{OPT}} \) elements. Then, the number of elements remaining after the first iteration is:

\[
n_1 \leq n - \frac{n}{\text{OPT}} = n \left( 1 - \frac{1}{\text{OPT}} \right)
\]

Some remaining set has at least \( \frac{n_1}{\text{OPT}} \) still not covered elements because?

Case 1: The first set we picked is not in the optimal solution.

Then, those \( n_1 \) elements must also be covered by OPT (i.e., same argument).

Case 2: The first set we picked is in the optimal solution.

ALG = # sets selected by the algorithm to cover all \( n \) elements.

OPT = # sets in an optimal solution to cover all \( n \) elements.
Set Cover – Performance

Suppose the universe contains $n$ elements.
The first set selected will have $\geq \frac{n}{\text{OPT}}$ elements.
Then, the number of elements remaining after the first iteration is:

$$n_1 \leq n - \frac{n}{\text{OPT}} = n \left(1 - \frac{1}{\text{OPT}}\right)$$

Some remaining set has at least $\frac{n_1}{\text{OPT}}$ still not covered elements because?

Case 1: The first set we picked is not in the optimal solution.

Then, those $n_1$ elements must also be covered by OPT (i.e., same argument).

Case 2: The first set we picked is in the optimal solution.

Then, a remaining set must have at least $\frac{n_1}{\text{OPT} - 1}$ uncovered elements,
Set Cover – Performance

Suppose the universe contains $n$ elements.
The first set selected will have $\geq \frac{n}{\text{OPT}}$ elements.
Then, the number of elements remaining after the first iteration is:

$$n_1 \leq n - \frac{n}{\text{OPT}} = n \left(1 - \frac{1}{\text{OPT}}\right)$$

Some remaining set has at least $\frac{n_1}{\text{OPT}}$ still not covered elements because?

Case 1: The first set we picked is not in the optimal solution.

Then, those $n_1$ elements must also be covered by OPT (i.e., same argument).

Case 2: The first set we picked is in the optimal solution.

Then, a remaining set must have at least $\frac{n_1}{\text{OPT} - 1}$ uncovered elements, which is larger, so at worst (smallest), the set has $\frac{n_1}{\text{OPT}}$ uncovered elements.
Set Cover – Performance

Suppose the universe contains \( n \) elements.
The first set selected will have \( \geq \frac{n}{\text{OPT}} \) elements.
Then, the number of elements remaining after the first iteration is:

\[
n_1 \leq n - \frac{n}{\text{OPT}} = n \left(1 - \frac{1}{\text{OPT}}\right)
\]

Some remaining set has at least \( \frac{n_1}{\text{OPT}} \) still not covered elements.
Then, the number of elements remaining after the second iteration is:

\[?\]
Set Cover – Performance

Suppose the universe contains $n$ elements. The first set selected will have $\geq \frac{n}{\text{OPT}}$ elements. Then, the number of elements remaining after the first iteration is:

$$n_1 \leq n - \frac{n}{\text{OPT}} = n \left(1 - \frac{1}{\text{OPT}}\right)$$

Some remaining set has at least $\frac{n_1}{\text{OPT}}$ still not covered elements. Then, the number of elements remaining after the second iteration is:

$$n_2 \leq n_1 - \frac{n_1}{\text{OPT}} = n_1 \left(1 - \frac{1}{\text{OPT}}\right) \leq n \left(1 - \frac{1}{\text{OPT}}\right)^2$$
Set Cover – Performance

Suppose the universe contains $n$ elements. The first set selected will have \( \geq \frac{n}{\text{OPT}} \) elements.

Then, the number of elements remaining after the first iteration is:

\[
    n_1 \leq n - \frac{n}{\text{OPT}} = n \left(1 - \frac{1}{\text{OPT}}\right)
\]

Some remaining set has at least \( \frac{n_1}{\text{OPT}} \) still not covered elements.

Then, the number of elements remaining after the second iteration is:

\[
    n_2 \leq n_1 - \frac{n_1}{\text{OPT}} = n_1 \left(1 - \frac{1}{\text{OPT}}\right) \leq n \left(1 - \frac{1}{\text{OPT}}\right)^2
\]

In general, after $t$ iterations,

\[
    n_t \leq n_{t-1} - \frac{n_{t-1}}{\text{OPT}} = n_{t-1} \left(1 - \frac{1}{\text{OPT}}\right) \leq n \left(1 - \frac{1}{\text{OPT}}\right)^t
\]
Set Cover – Performance

Suppose the universe contains $n$ elements. Before the $t^{\text{th}}$ iteration, some remaining set has at least $\frac{n_{t-1}}{\text{OPT}}$ uncovered elements and the number of elements remaining after the $t^{\text{th}}$ iteration is:

$$n_t \leq n_{t-1} - \frac{n_{t-1}}{\text{OPT}} = n_{t-1} \left(1 - \frac{1}{\text{OPT}}\right) \leq n \left(1 - \frac{1}{\text{OPT}}\right)^t$$

ALG = # sets selected by the algorithm to cover all $n$ elements.
OPT = # sets in an optimal solution to cover all $n$ elements.
Set Cover – Performance

Suppose the universe contains $n$ elements. Before the $t^{th}$ iteration, some remaining set has at least $\frac{n_{t-1}}{\text{OPT}}$ uncovered elements and the number of elements remaining after the $t^{th}$ iteration is:

$$n_t \leq n_{t-1} - \frac{n_{t-1}}{\text{OPT}} = n_{t-1} \left(1 - \frac{1}{\text{OPT}}\right) \leq n \left(1 - \frac{1}{\text{OPT}}\right)^t$$

Big picture:
Set Cover – Performance

Suppose the universe contains $n$ elements. Before the $t$th iteration, some remaining set has at least $\frac{n_{t-1}}{\text{OPT}}$ uncovered elements and the number of elements remaining after the $t$th iteration is:

$$n_t \leq n_{t-1} - \frac{n_{t-1}}{\text{OPT}} = n_{t-1} \left(1 - \frac{1}{\text{OPT}}\right) \leq n \left(1 - \frac{1}{\text{OPT}}\right)^t$$

Big picture:
How many sets do I add each iteration?
Set Cover – Performance

Suppose the universe contains $n$ elements.
Before the $t^{th}$ iteration, some remaining set has at least $\frac{n_{t-1}}{\text{OPT}}$ uncovered elements and the number of elements remaining after the $t^{th}$ iteration is:

$$n_t \leq n_t-1 - \frac{n_{t-1}}{\text{OPT}} = n_{t-1} \left(1 - \frac{1}{\text{OPT}}\right) \leq n \left(1 - \frac{1}{\text{OPT}}\right)^t$$

Big picture:
How many sets do I add each iteration? 1

ALG = # sets selected by the algorithm to cover all $n$ elements.
OPT = # sets in an optimal solution to cover all $n$ elements.
Set Cover – Performance

Suppose the universe contains $n$ elements. Before the $t^{th}$ iteration, some remaining set has at least $\frac{n_{t-1}}{\text{OPT}}$ uncovered elements and the number of elements remaining after the $t^{th}$ iteration is:

$$n_t \leq n_{t-1} - \frac{n_{t-1}}{\text{OPT}} = n_{t-1} \left(1 - \frac{1}{\text{OPT}}\right) \leq n \left(1 - \frac{1}{\text{OPT}}\right)^t$$

Big picture:
How many sets do I add each iteration? 1
$\text{ALG} = ?$
Set Cover – Performance

Suppose the universe contains $n$ elements. Before the $t$th iteration, some remaining set has at least $\frac{n_{t-1}}{\text{OPT}}$ uncovered elements and the number of elements remaining after the $t$th iteration is:

$$n_t \leq n_{t-1} - \frac{n_{t-1}}{\text{OPT}} = n_{t-1} \left(1 - \frac{1}{\text{OPT}}\right) \leq n \left(1 - \frac{1}{\text{OPT}}\right)^t$$

Big picture:
How many sets do I add each iteration? 1
ALG = # iterations
Set Cover – Performance

Suppose the universe contains $n$ elements.
Before the $t^{th}$ iteration, some remaining set has at least $\frac{n_{t-1}}{\text{OPT}}$ uncovered elements and the number of elements remaining after the $t^{th}$ iteration is:

$$n_t \leq n_{t-1} - \frac{n_{t-1}}{\text{OPT}} = n_{t-1} \left(1 - \frac{1}{\text{OPT}}\right) \leq n \left(1 - \frac{1}{\text{OPT}}\right)^t$$

Big picture:
How many sets do I add each iteration? 1
ALG = # iterations
How many iterations until $n_t < 1$?
Set Cover – Performance

Suppose the universe contains $n$ elements. Before the $t^{th}$ iteration, some remaining set has at least $\frac{n_{t-1}}{\text{OPT}}$ uncovered elements and the number of elements remaining after the $t^{th}$ iteration is:

$$n_t \leq n_{t-1} - \frac{n_{t-1}}{\text{OPT}} = n_{t-1} \left(1 - \frac{1}{\text{OPT}}\right) \leq n \left(1 - \frac{1}{\text{OPT}}\right)^t$$

Accepting that $1 - x < e^{-x}$ for all $x \neq 0$,

$$n_t \leq ?$$

ALG = # sets selected by the algorithm to cover all $n$ elements.
OPT = # sets in an optimal solution to cover all $n$ elements.
Set Cover – Performance

Suppose the universe contains \( n \) elements.

Before the \( t^{th} \) iteration, some remaining set has at least \( \frac{n_{t-1}}{OPT} \) uncovered elements and the number of elements remaining after the \( t^{th} \) iteration is:

\[
n_t \leq n_{t-1} - \frac{n_{t-1}}{OPT} = n_{t-1} \left(1 - \frac{1}{OPT}\right) \leq n \left(1 - \frac{1}{OPT}\right)^t
\]

Accepting that \( 1 - x < e^{-x} \) for all \( x \neq 0 \),

\[
n_t \leq n \left(1 - \frac{1}{OPT}\right)^t < n \left(e^{-\frac{1}{OPT}}\right)^t = ne^{-\frac{t}{OPT}}
\]
Suppose the universe contains $n$ elements.
Before the $t^{th}$ iteration, some remaining set has at least $\frac{n_{t-1}}{OPT}$ uncovered elements and the number of elements remaining after the $t^{th}$ iteration is:

$$n_t \leq n_{t-1} - \frac{n_{t-1}}{OPT} = n_{t-1} \left(1 - \frac{1}{OPT}\right) \leq n \left(1 - \frac{1}{OPT}\right)^t$$

Accepting that $1 - x < e^{-x}$ for all $x \neq 0$,

$$n_t \leq n \left(1 - \frac{1}{OPT}\right)^t < n \left(e^{-\frac{1}{OPT}}\right)^t = n e^{-\frac{t}{OPT}}$$

What does $t$ equal to make $n e^{-\frac{t}{OPT}} < 1$?
Suppose the universe contains $n$ elements. Before the $t^{th}$ iteration, some remaining set has at least $\frac{n_{t-1}}{\text{OPT}}$ uncovered elements and the number of elements remaining after the $t^{th}$ iteration is:

$$n_t \leq n_{t-1} - \frac{n_{t-1}}{\text{OPT}} = n_{t-1} \left(1 - \frac{1}{\text{OPT}}\right) \leq n \left(1 - \frac{1}{\text{OPT}}\right)^t$$

Accepting that $1 - x < e^{-x}$ for all $x \neq 0$,

$$n_t \leq n \left(1 - \frac{1}{\text{OPT}}\right)^t < n \left(e^{-\frac{1}{\text{OPT}}}\right)^t = ne^{-\frac{t}{\text{OPT}}}$$

If $t = \text{OPT} \ln n$, $n_t < ?$
Set Cover – Performance

Suppose the universe contains \( n \) elements.
Before the \( t^{th} \) iteration, some remaining set has at least \( \frac{n_{t-1}}{\text{OPT}} \) uncovered elements and the number of elements remaining after the \( t^{th} \) iteration is:

\[
n_t \leq n_{t-1} - \frac{n_{t-1}}{\text{OPT}} = n_{t-1} \left( 1 - \frac{1}{\text{OPT}} \right) \leq n \left( 1 - \frac{1}{\text{OPT}} \right)^t
\]

Accepting that \( 1 - x < e^{-x} \) for all \( x \neq 0 \),

\[
n_t \leq n \left( 1 - \frac{1}{\text{OPT}} \right)^t < n \left( e^{-\frac{1}{\text{OPT}}} \right)^t = ne^{-\frac{t}{\text{OPT}}}
\]

If \( t = \text{OPT} \ln n \), \( n_t < ne^{-\frac{\text{OPT} \ln n}{\text{OPT}}} = 1 \)
Set Cover – Performance

Suppose the universe contains $n$ elements. Before the $t^{th}$ iteration, some remaining set has at least $\frac{n_{t-1}}{\text{OPT}}$ uncovered elements and the number of elements remaining after the $t^{th}$ iteration is:

$$n_t \leq n_{t-1} - \frac{n_{t-1}}{\text{OPT}} = n_{t-1} \left(1 - \frac{1}{\text{OPT}}\right) \leq n \left(1 - \frac{1}{\text{OPT}}\right)^t$$

Accepting that $1 - x < e^{-x}$ for all $x \neq 0$,

$$n_t \leq n \left(1 - \frac{1}{\text{OPT}}\right)^t < n \left(e^{-\frac{1}{\text{OPT}}}\right)^t = ne^{-\frac{t}{\text{OPT}}}$$

If $t = \text{OPT} \ln n$, $n_t < ne^{-\frac{\text{OPT} \ln n}{\text{OPT}}} = 1$, which means that no elements remain.
Set Cover – Performance

Suppose the universe contains $n$ elements.

Before the $t^{th}$ iteration, some remaining set has at least $\frac{n_{t-1}}{OPT}$ uncovered elements and the number of elements remaining after the $t^{th}$ iteration is:

$$n_t \leq n_{t-1} - \frac{n_{t-1}}{OPT} = n_{t-1} \left(1 - \frac{1}{OPT}\right) \leq n \left(1 - \frac{1}{OPT}\right)^t$$

Accepting that $1 - x < e^{-x}$ for all $x \neq 0$,

$$n_t \leq n \left(1 - \frac{1}{OPT}\right)^t < n \left(e^{-\frac{1}{OPT}}\right)^t = ne^{-t/OPT}$$

If $t = OPT \ln n$, $n_t < ne^{-OPT \ln n/OPT} = 1$, which means that no elements remain. So, the universe is covered after at most $t = OPT \ln n$ iterations.
Set Cover – Performance

Suppose the universe contains \( n \) elements.

Before the \( t^{\text{th}} \) iteration, some remaining set has at least \( \frac{n_{t-1}}{\text{OPT}} \) uncovered elements and the number of elements remaining after the \( t^{\text{th}} \) iteration is:

\[
n_t \leq n_{t-1} - \frac{n_{t-1}}{\text{OPT}} = n_{t-1} \left(1 - \frac{1}{\text{OPT}}\right) \leq n \left(1 - \frac{1}{\text{OPT}}\right)^t
\]

Accepting that \( 1 - x < e^{-x} \) for all \( x \neq 0 \),

\[
n_t \leq n \left(1 - \frac{1}{\text{OPT}}\right)^t < n \left(e^{-\frac{1}{\text{OPT}}}\right)^t = ne^{-\frac{t}{\text{OPT}}}
\]

If \( t = \text{OPT} \ln n \), then
\[
n_t < ne^{-\frac{\text{OPT} \ln n}{\text{OPT}}} = 1,
\]
which means that no elements remain. So, the universe is covered after at most \( t = \text{OPT} \ln n \) iterations.

\[
\Rightarrow \text{ALG} \leq \ln n \text{ OPT}
\]

ALG = # sets selected by the algorithm to cover all \( n \) elements.

OPT = # sets in an optimal solution to cover all \( n \) elements.
Set Cover – Tightness

Find an instance of 16 elements where the optimal solution is 2, but the algorithm will find a solution of 4.
Set Cover – Tightness

Find an instance of 16 elements where the optimal solution is 2, but the algorithm will find a solution of 4.
Set Cover – Tightness

Find an instance of 16 elements where the optimal solution is 2, but the algorithm will find a solution of 4.

In General:

Universe of size $n = 2^k$. OPT = 2. ALG = $k$.

$\Rightarrow k \in \Omega(\log_2 2^k) = \Omega(\ln 2^k) = \Omega(\ln n)$
Set Cover – Inapproximability

It turns out that Set Cover cannot be approximated within the bound of \((1 - o(1)) \ln n\), unless P = NP.
Set Cover – Inapproximability

It turns out that Set Cover cannot be approximated within the bound of \( (1 - o(1)) \ln n \), unless \( P = NP \).

APX: Set of optimization problems that can be approximated within a constant ratio.

Vertex Cover \( \in \) APX
Set Cover \( \notin \) APX