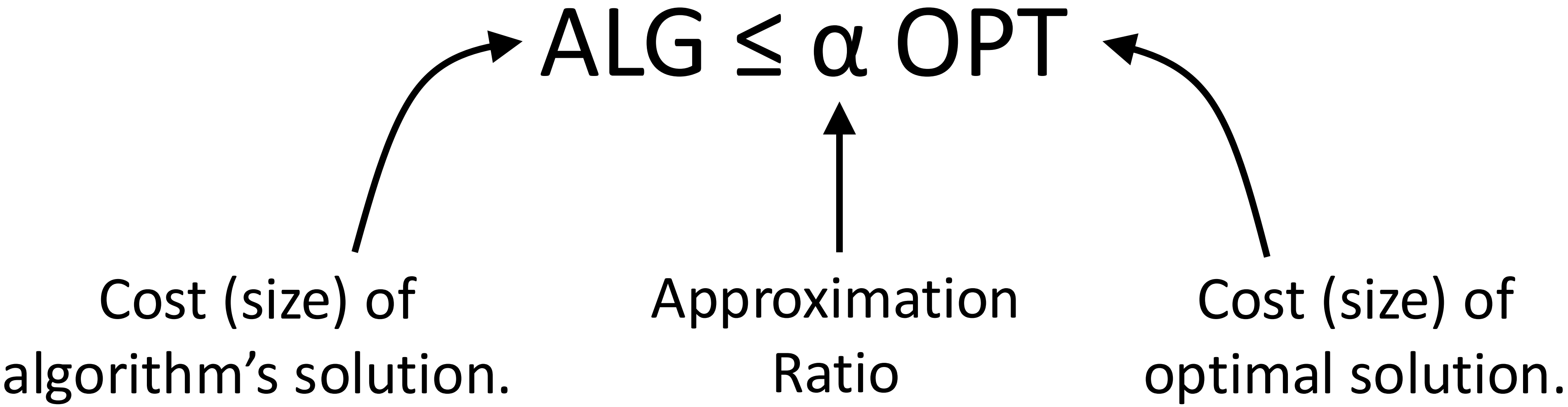
Approximation Algorithms
CSCI 532

Approximation Algorithms



$$\text{ALG} \le \alpha \ \text{OPT}$$

Cost (size) of algorithm's solution.

Approximation Ratio

Cost (size) of optimal solution.

if problem is a maximization problem, $\text{ALG} \ge \frac{1}{\alpha} \text{OPT}$

Vertex Cover

```
while uncovered edge exists
    select both of its vertices
```

Consider a set of edges, $E' \subset E$, that do not share vertices. Is there a relationship between the minimum vertex cover and $|E'|$?

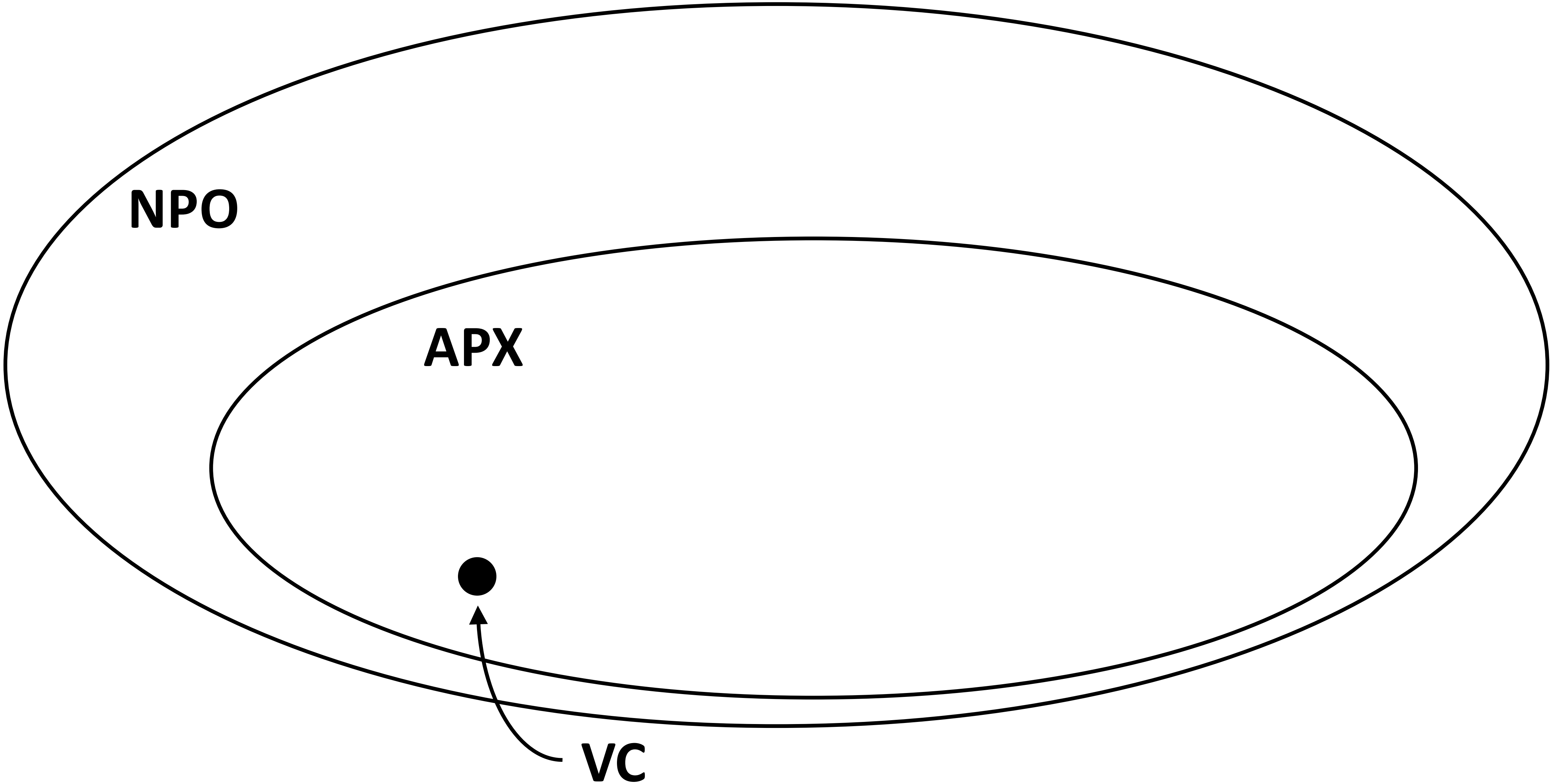
$$|E'| \leq \text{OPT}$$

Does the size of the algorithm's output relate to a set of edges that do not share vertices?

$$\text{ALG} = 2\,|E'|$$

$$\Longrightarrow \text{ALG} = 2\,|E'| \leq 2\ \text{OPT} \Longrightarrow \text{ALG} \leq 2\ \text{OPT}$$

Approximability Hierarchy



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:

$$U = \{1, 4, 7, 8, 10\}$$

$$S = \{\{1, 7, 8\}, \{1, 4, 7\}, \{7, 8\}, \{4, 8, 10\}\}$$

$\{\{1, 7, 8\}, \{4, 8, 10\}\}$ ✓

$\{\{1, 4, 7\}, \{7, 8\}\}$ ✗

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.

Greedy Algorithm:

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

Set Cover

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.

$$ALG \leq \alpha\, OPT$$

Game Plan:

Bound the maximum number of sets in ALG…

By bounding the maximum number of iterations of the algorithm...

By bounding the size of each set added by the algorithm.

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

OPT = # sets in an optimal solution to cover all n elements.

Set Cover

Suppose the universe contains n elements.

What can we say about the first set selected?

It’s the biggest!

? $\leq$ |Biggest Set| $\leq$?

Guarantee we do at least this good.

Guarantee we don’t do better than this.

Set Cover

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.

What can we say about the first set selected?

It’s the biggest!

Claim: $\boldsymbol{\frac{n}{OPT}} \leq$ **|Biggest Set|**

Why?

Set Cover

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.

What can we say about the first set selected?

It's the biggest!

Claim: $\boldsymbol{\frac{n}{OPT} \leq}$ **|Biggest Set|**

Suppose $\frac{n}{OPT} >$ |Biggest Set|

Set Cover

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.

What can we say about the first set selected?

It's the biggest!

Claim: $\boldsymbol{\frac{n}{OPT} \leq |\textbf{Biggest Set}|}$

Suppose $\frac{n}{OPT} > |\text{Biggest Set}| \geq |\text{Every Set}|$

Set Cover

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.

What can we say about the first set selected?

It's the biggest!

Claim: $\boldsymbol{\frac{n}{OPT}} \leq$ **|Biggest Set|**

Suppose $\frac{n}{OPT} >$ |Biggest Set| $\geq$ |Every Set|

How many sets does the optimal solution S^{OPT} use?

Set Cover

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.

What can we say about the first set selected?

It's the biggest!

Claim: $\boldsymbol{\frac{n}{OPT}} \leq$ **|Biggest Set|**

Suppose $\frac{n}{OPT} >$ |Biggest Set| $\geq$ |Every Set|

How many sets does the optimal solution S^{OPT} use? $\boldsymbol{OPT}$

Set Cover

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.

What can we say about the first set selected?

It’s the biggest!

Claim: $\boldsymbol{\frac{n}{OPT} \leq |\text{Biggest Set}|}$

Suppose $\frac{n}{OPT} > |\text{Biggest Set}| \geq |\text{Every Set}|$

How many sets does the optimal solution S^{OPT} use? OPT

Then how many elements are covered by those OPT sets?

Set Cover

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.

What can we say about the first set selected?

It’s the biggest!

Claim: $\boldsymbol{\frac{n}{OPT} \leq}$ **|Biggest Set|**

Suppose $\frac{n}{OPT} >$ |Biggest Set| $\geq$ |Every Set|

How many sets does the optimal solution S^{OPT} use? OPT

Then how many elements are covered by those OPT sets?

$$\begin{pmatrix}\text{\# elements}\\ \text{covered by}\\ S^{OPT}\end{pmatrix} \leq \sum_{S \in S^{OPT}} |S|$$

Elements in S^{OPT} may be repeated

Set Cover

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.

What can we say about the first set selected?

It's the biggest!

Claim: $\boldsymbol{\frac{n}{OPT} \leq |\text{Biggest Set}|}$

Suppose $\frac{n}{OPT} > |\text{Biggest Set}| \geq |\text{Every Set}|$

How many sets does the optimal solution S^{OPT} use? OPT

Then how many elements are covered by those OPT sets?

$$\begin{pmatrix}\text{\# elements}\\ \text{covered by}\\ S^{OPT}\end{pmatrix} \leq \sum_{S \in S^{OPT}} |S| \leq \sum_{S \in S^{OPT}} |\text{Biggest Set}|$$

Set Cover

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.

What can we say about the first set selected?

It's the biggest!

Claim: $\boldsymbol{\frac{n}{OPT} \leq}$ **|Biggest Set|**

Suppose $\frac{n}{OPT} > |\text{Biggest Set}| \geq |\text{Every Set}|$

How many sets does the optimal solution S^{OPT} use? OPT

Then how many elements are covered by those OPT sets?

$$\begin{pmatrix}\text{\# elements} \\ \text{covered by} \\ S^{OPT}\end{pmatrix} \leq \sum_{S \in S^{OPT}} |S| \leq \sum_{S \in S^{OPT}} |\text{Biggest Set}| = OPT|\text{Biggest Set}|$$

Set Cover

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.

What can we say about the first set selected?

It's the biggest!

Claim: $\boldsymbol{\frac{n}{OPT} \leq |\textbf{Biggest Set}|}$

Suppose $\boxed{\frac{n}{OPT} > |\text{Biggest Set}|} \geq |\text{Every Set}|$

How many sets does the optimal solution S^{OPT} use? OPT

Then how many elements are covered by those OPT sets?

$$\begin{pmatrix}\text{\# elements} \\ \text{covered by} \\ S^{OPT}\end{pmatrix} \leq \sum_{S \in S^{OPT}} |S| \leq \sum_{S \in S^{OPT}} |\text{Biggest Set}| = OPT|\text{Biggest Set}| < OPT\frac{n}{OPT}$$

Set Cover

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.

What can we say about the first set selected?

It's the biggest!

Claim: $\boldsymbol{\frac{n}{OPT} \leq |\text{Biggest Set}|}$

Suppose $\frac{n}{OPT} > |\text{Biggest Set}| \geq |\text{Every Set}|$

How many sets does the optimal solution S^{OPT} use? OPT

Then how many elements are covered by those OPT sets?

$$\begin{pmatrix}\text{\# elements} \\ \text{covered by} \\ S^{OPT}\end{pmatrix} \leq \sum_{S \in S^{OPT}} |S| \leq \sum_{S \in S^{OPT}} |\text{Biggest Set}| = OPT|\text{Biggest Set}| < OPT\frac{n}{OPT} = n$$

Set Cover

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.

What can we say about the first set selected?

It's the biggest!

Claim: $\boldsymbol{\frac{n}{OPT}} \leq$ **|Biggest Set|**

Suppose $\frac{n}{OPT} >$ |Biggest Set| $\geq$ |Every Set|

How many sets does the optimal solution S^{OPT} use? OPT

Then how many elements are covered by those OPT sets?

$$\left(\begin{array}{c}\textbf{\# elements}\\ \textbf{covered by}\\ \boldsymbol{S^{OPT}}\end{array}\right) \leq \sum_{S \in S^{OPT}} |S| \leq \sum_{S \in S^{OPT}} |\text{Biggest Set}| = OPT\,|\text{Biggest Set}| < OPT\,\frac{n}{OPT} = \boldsymbol{n}$$

$\boldsymbol{S^{OPT}}$ doesn't cover all $\boldsymbol{n}$ elements

Set Cover

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.

What can we say about the first set selected?

It’s the biggest!

Claim: $\boldsymbol{\frac{n}{OPT} \leq |\text{Biggest Set}|}$

Suppose $\frac{n}{OPT} > |\text{Biggest Set}| \geq |\text{Every Set}|$

How many sets does the optimal solution S^{OPT} use? OPT

Then how many elements are covered by those OPT sets?

$$\left(\begin{array}{c}\textbf{\# elements}\\ \textbf{covered by}\\ \boldsymbol{S^{OPT}}\end{array}\right) \leq \sum_{S \in S^{OPT}} |S| \leq \sum_{S \in S^{OPT}} |\text{Biggest Set}| = OPT|\text{Biggest Set}| < OPT\frac{n}{OPT} = \boldsymbol{n}$$

S^{OPT} doesn’t cover all n elements

Set Cover

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 $\geq \frac{n}{OPT}$ elements.

Set Cover

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 $\geq \frac{n}{OPT}$ elements.
Then, the number of elements remaining after the first iteration is:

Original # elements $-$ **# elements removed**

$\boldsymbol{n}$

$\geq \frac{\boldsymbol{n}}{\boldsymbol{OPT}}$

Set Cover

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 $\geq \frac{n}{OPT}$ elements.
Then, the number of elements remaining after the first iteration is:

$$n_1 \leq n - \frac{n}{OPT}$$

The first set could be $> \frac{n}{OPT}$ which would leave fewer elements remaining.

Original # elements − **# elements removed**

$$\mathbf{n} \qquad \geq \frac{\mathbf{n}}{\mathbf{OPT}}$$

Set Cover

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 $\geq \frac{n}{OPT}$ elements.

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

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

Set Cover

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 $\geq \frac{n}{OPT}$ elements.

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

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

What can we say about the second set selected?

Set Cover

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 $\geq \frac{n}{OPT}$ elements.

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

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

What can we say about the second set selected? ⟵ **It covers the most uncovered elements.**

Set Cover

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 $\geq \frac{n}{OPT}$ elements.

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

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

What can we say about the second set selected?

Suppose the first set <u>was</u> in the optimal solution.

Suppose the first set <u>was not</u> in the optimal solution.

Set Cover

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 $\geq \frac{n}{OPT}$ elements.
Then, the number of elements remaining after the first iteration is:

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

What can we say about the second set selected?

Suppose the first set was in the optimal solution.

Then, a remaining set must have at least $\frac{n_1}{OPT-1}$ uncovered elements.

Suppose the first set was not in the optimal solution.

If not, how do the remaining $OPT - 1$ optimal sets cover the remaining n_1 elements?

Set Cover

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 $\geq \frac{n}{OPT}$ elements.

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

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

What can we say about the second set selected?

Suppose the first set <u>was</u> in the optimal solution.

Then, a remaining set must have at least $\frac{n_1}{OPT-1}$ uncovered elements.

$$\Rightarrow \frac{n_1}{OPT-1} \leq |\text{Second Set}|$$

Suppose the first set <u>was not</u> in the optimal solution.

Set Cover

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 $\geq \frac{n}{OPT}$ elements.
Then, the number of elements remaining after the first iteration is:

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

What can we say about the second set selected?

Suppose the first set <u>was</u> in the optimal solution.

Then, a remaining set must have at least $\frac{n_1}{OPT-1}$ uncovered elements.

$$\Rightarrow \frac{n_1}{OPT-1} \leq |\text{Second Set}|$$

Suppose the first set <u>was not</u> in the optimal solution.

Then, the n_1 elements must still be covered by OPT.

Set Cover

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 $\geq \frac{n}{OPT}$ elements.

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

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

What can we say about the second set selected?

Suppose the first set was in the optimal solution.

Then, a remaining set must have at least $\frac{n_1}{OPT-1}$ uncovered elements.

$$\Rightarrow \frac{n_1}{OPT-1} \leq |\text{Second Set}|$$

Suppose the first set was not in the optimal solution.

Then, the n_1 elements must still be covered by OPT.

$$\Rightarrow \frac{n_1}{OPT} \leq |\text{Second Set}|$$

If not, how do the OPT optimal sets cover the remaining n_1 elements?

Set Cover

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 $\geq \frac{n}{OPT}$ elements.
Then, the number of elements remaining after the first iteration is:

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

What can we say about the second set selected?

Suppose the first set was in the optimal solution.

Then, a remaining set must have at least $\frac{n_1}{OPT-1}$ uncovered elements.

$$\Rightarrow \frac{n_1}{OPT} < \frac{n_1}{OPT-1} \leq |\text{Second Set}|$$

Suppose the first set was not in the optimal solution.

Then, the n_1 elements must still be covered by OPT.

$$\Rightarrow \frac{n_1}{OPT} \leq |\text{Second Set}|$$

Set Cover

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 $\geq \frac{n}{OPT}$ elements.
Then, the number of elements remaining after the first iteration is:

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

Some remaining set has at least $\frac{n_1}{OPT}$ uncovered elements.

Suppose the first set was in the optimal solution.

Then, a remaining set must have at least $\frac{n_1}{OPT-1}$ uncovered elements.

$$\Rightarrow \frac{n_1}{OPT} < \frac{n_1}{OPT-1} \leq |\text{Second Set}|$$

Suppose the first set was not in the optimal solution.

Then, the n_1 elements must still be covered by OPT.

$$\Rightarrow \frac{n_1}{OPT} \leq |\text{Second Set}|$$

Set Cover

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 $\geq \frac{n}{OPT}$ elements.

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

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

Some remaining set has at least $\frac{n_1}{OPT}$ uncovered elements.

Set Cover

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 $\geq \frac{n}{OPT}$ elements.
Then, the number of elements remaining after the first iteration is:

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

Some remaining set has at least $\frac{n_1}{OPT}$ uncovered elements.
Then, the number of elements remaining after the second iteration is:

Set Cover

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 $\geq \frac{n}{OPT}$ elements.

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

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

Some remaining set has at least $\frac{n_1}{OPT}$ uncovered elements.

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

$$n_2 \leq n_1 - \frac{n_1}{OPT}$$

Set Cover

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 $\geq \frac{n}{OPT}$ elements.

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

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

Some remaining set has at least $\frac{n_1}{OPT}$ uncovered elements.

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

$$n_2 \leq n_1 - \frac{n_1}{OPT} = n_1\left(1 - \frac{1}{OPT}\right)$$

Set Cover

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 $\geq \frac{n}{OPT}$ elements.

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

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

Some remaining set has at least $\frac{n_1}{OPT}$ uncovered elements.

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

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

Set Cover

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 $\geq \frac{n}{OPT}$ elements.
Then, the number of elements remaining after the first iteration is:

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

Some remaining set has at least $\frac{n_1}{OPT}$ uncovered elements.
Then, the number of elements remaining after the second iteration is:

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

In general, after t iterations:

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

Set Cover

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.
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:

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

Set Cover

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.
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:

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

Big picture:

How many sets are added each iteration?

Set Cover

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.
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:

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

Big picture:

How many sets are added each iteration? **1**

Set Cover

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.
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:

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

Big picture:

How many sets are added each iteration? **1**

ALG = **# iterations**

Set Cover

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.
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:

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

Big picture:

How many sets are added each iteration? **1**

ALG = **# iterations**

When does the algorithm terminate?

Set Cover

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.
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:

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

Big picture:

How many sets are added each iteration? **1**

ALG = **# iterations**

When does the algorithm terminate? **When $\boldsymbol{n_t < 1}$**

Set Cover

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.
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:

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

Big picture:

How many sets are added each iteration? **1**

ALG = **# iterations**

When does the algorithm terminate? **When $n_t < 1$**

What value of t makes $n_t < 1$?

Set Cover

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.
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:

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

Trust that: $1 - x < e^{-x}$ for all $x \neq 0$

What value of t makes $n_t < 1$?

Set Cover

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.

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:

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

Trust that: $1 - x < e^{-x}$ for all $x \neq 0$

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

What value of t makes $n_t < 1$?

Set Cover

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.
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:

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

Trust that: $1 - x < e^{-x}$ for all $x \neq 0$

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

What value of t makes $n_t < 1$?

Set Cover

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.
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:

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

Trust that: $1 - x < e^{-x}$ for all $x \neq 0$

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

$ne^{-\frac{t}{OPT}} \le 1 \Rightarrow t \ge OPT \ln n$

What value of t makes $n_t < 1$?

Set Cover

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.
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:

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

Trust that: $1 - x < e^{-x}$ for all $x \neq 0$

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

$ne^{-\frac{t}{OPT}} \leq 1 \Rightarrow t \geq OPT \ln n$

So, when $t = OPT \ln n$, $n_t < 1$ (i.e., no elements remain).

Set Cover

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.

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:

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

Trust that: $1 - x < e^{-x}$ for all $x \neq 0$

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

It could take fewer iterations, we just know that it can't take more.

$ne^{-\frac{t}{OPT}} \leq 1 \Rightarrow t \geq OPT \ln n$

So, when $t = OPT \ln n$, $n_t < 1$ (i.e., no elements remain). Thus, the universe is covered after at most $t = OPT \ln n$ iterations.

Set Cover

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.
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:

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

Trust that: $1 - x < e^{-x}$ for all $x \neq 0$

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

$ne^{-\frac{t}{OPT}} \leq 1 \Rightarrow t \geq OPT \ln n$

So, when $t = OPT \ln n$, $n_t < 1$ (i.e., no elements remain). Thus, the universe is covered after at most $t = OPT \ln n$ iterations.

$$\boldsymbol{\Longrightarrow ALG \leq \ln n \, OPT}$$

Set Cover

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.
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:

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

Trust that: $1 - x < e^{-x}$ for all $x \neq 0$

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

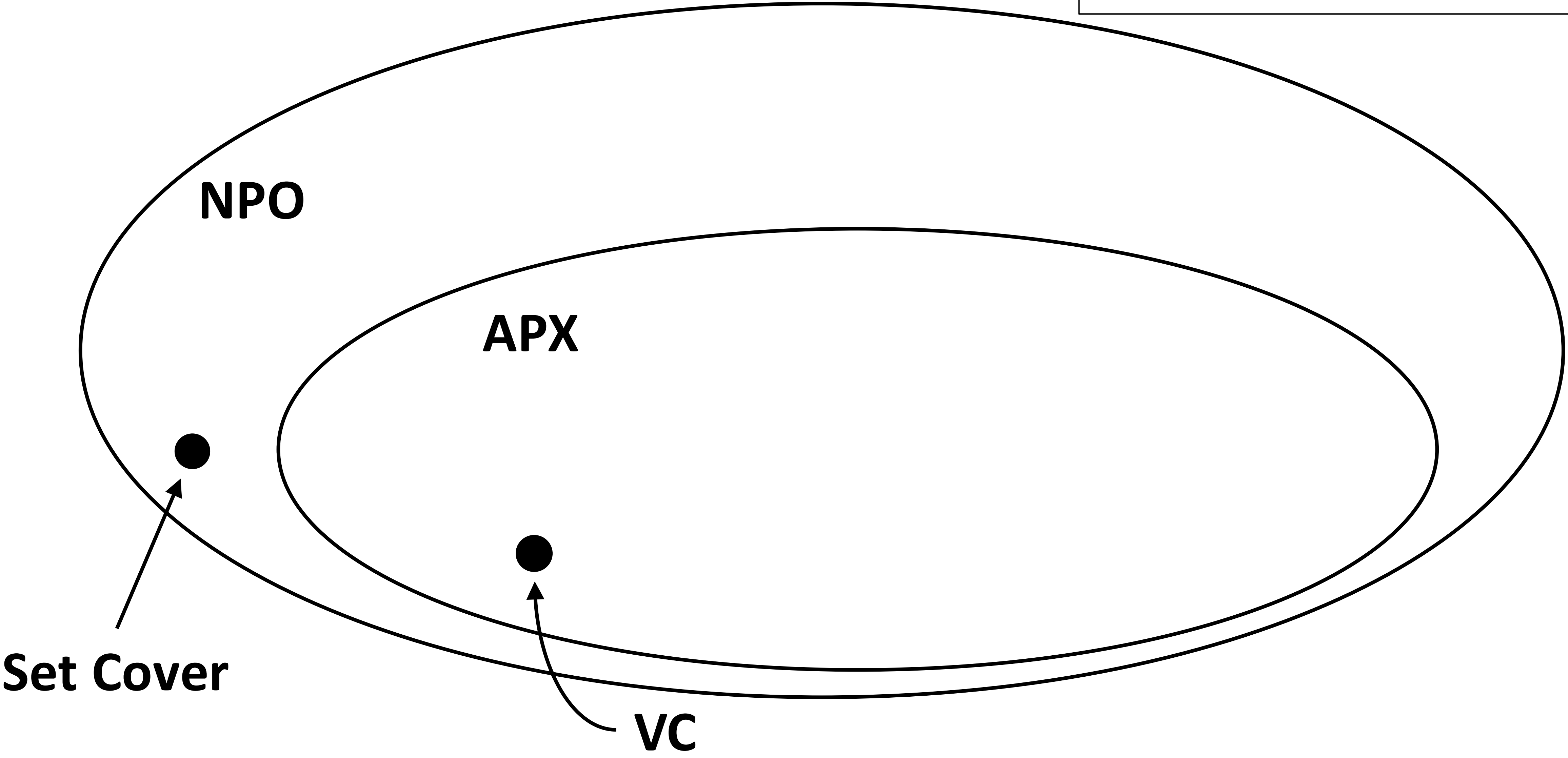
$$ne^{-\frac{t}{OPT}} \leq 1 \Rightarrow t \geq OPT \ln n$$

So, when $t = OPT \ln n$, $n_t < 1$ (i.e., no elements remain). Thus, the universe is covered after at most $t = OPT \ln n$ iterations.

$$\Longrightarrow \boldsymbol{ALG \leq \ln n\, OPT}$$

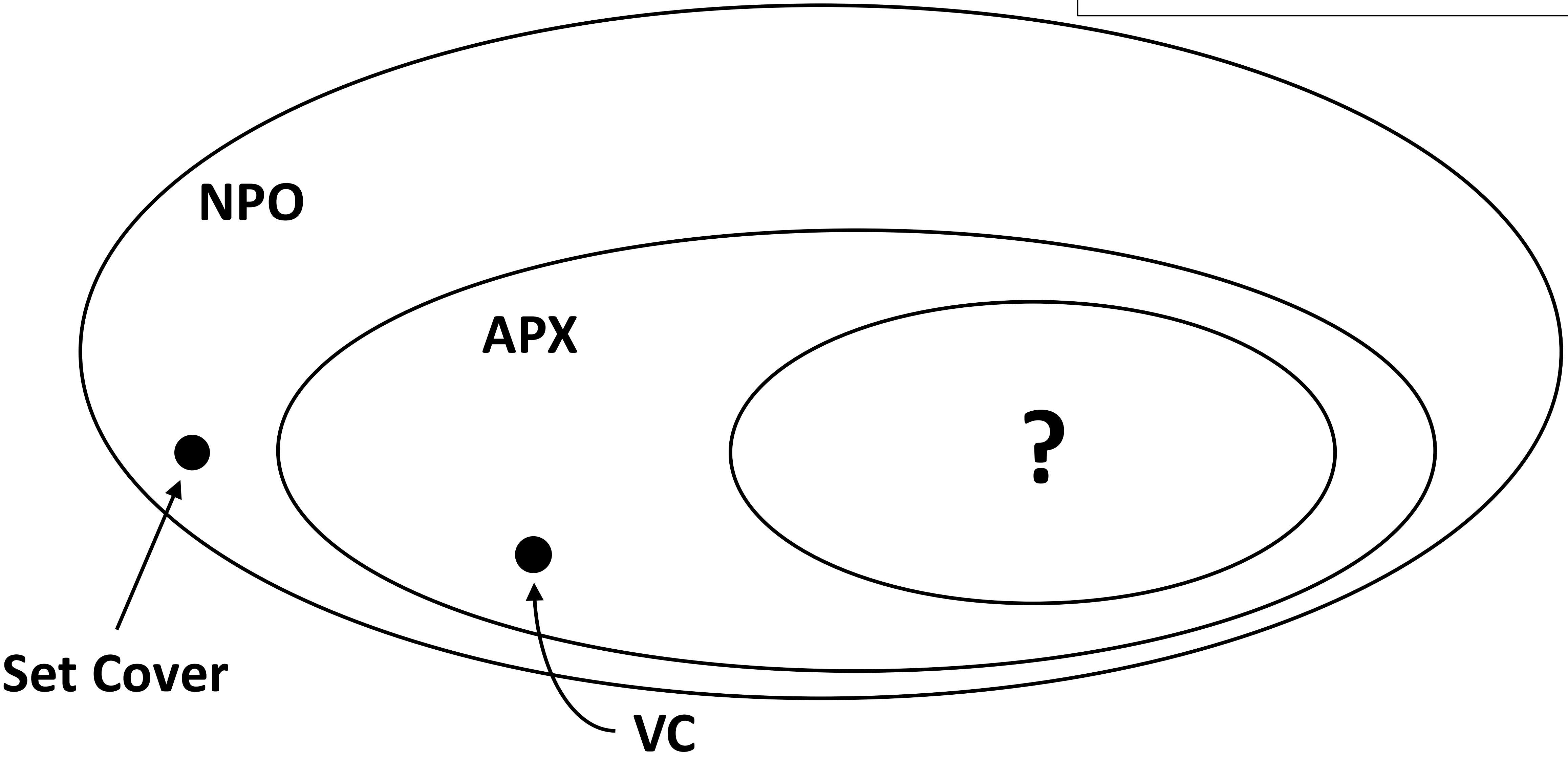
Approximability Hierarchy

APX: Optimization problems that can be approximated within a constant ratio.



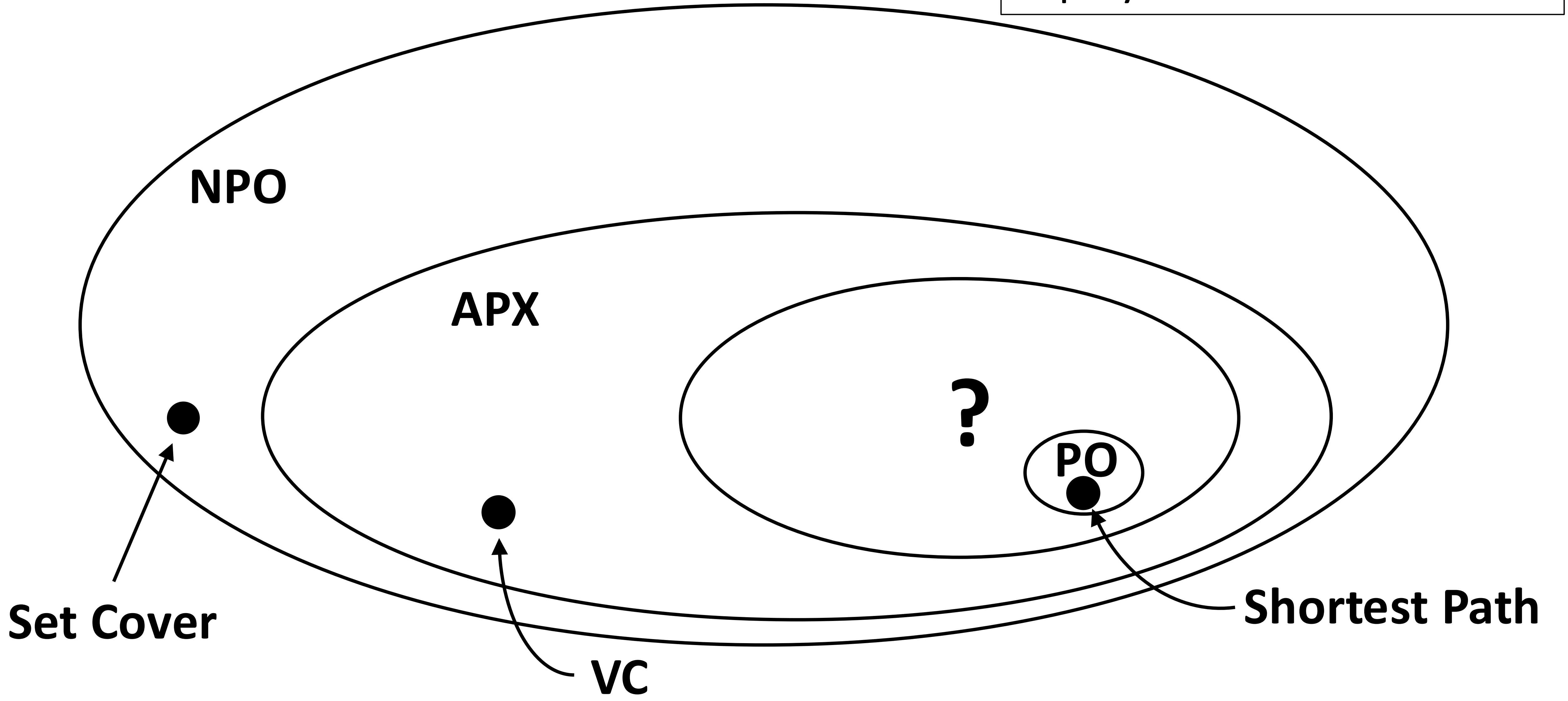
Approximability Hierarchy

APX: Optimization problems that can be approximated within a constant ratio.



Approximability Hierarchy

PO: Optimization problems that can be optimally solved in polynomial time.



Knapsack

Knapsack: Given a set of n items with values $v_1, \ldots, v_n$ and weights $w_1, \ldots, w_n$, select the most valuable combination with total weight $\leq W$.

Example:

$W = 11$

Item	Value	Weight
1	1	1
2	6	2
3	18	5
4	22	6
5	28	7

Knapsack

Knapsack: Given a set of n items with values $v_1, \dots, v_n$ and weights $w_1, \dots, w_n$, select the most valuable combination with total weight $\leq W$.

Example:

$W = 11$

Item	Value	Weight
1	1	1
2	6	2
3	18	5
4	22	6
5	28	7

$\{1, 2, 5\}$: weight = 10, value = 35

$\{3, 4\}$: weight = 11, value = 40

✓

$\{4, 5\}$: weight = 13, value = 50 ✗

Knapsack

Knapsack: Given a set of n items with values $v_1, \dots, v_n$ and weights $w_1, \dots, w_n$, select the most valuable combination with total weight $\leq W$.

Algorithm:

How could we approach this?

- Greedy?
- ILP?
- Flow Network?
- Dynamic Program?

Item	Value	Weight
1	11	6
2	11	6
3	20	10

$$W = 12$$

Knapsack – Dynamic Programming

Knapsack: Given a set of n items with values $v_1, \dots, v_n$ and weights $w_1, \dots, w_n$, select the most valuable combination with total weight $\leq W$.

Algorithm:

- Is there optimal substructure?
  - I.e. If I have an optimal solution to some instance, does that imply an optimal solution to a different instance?