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:

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

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

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:

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

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

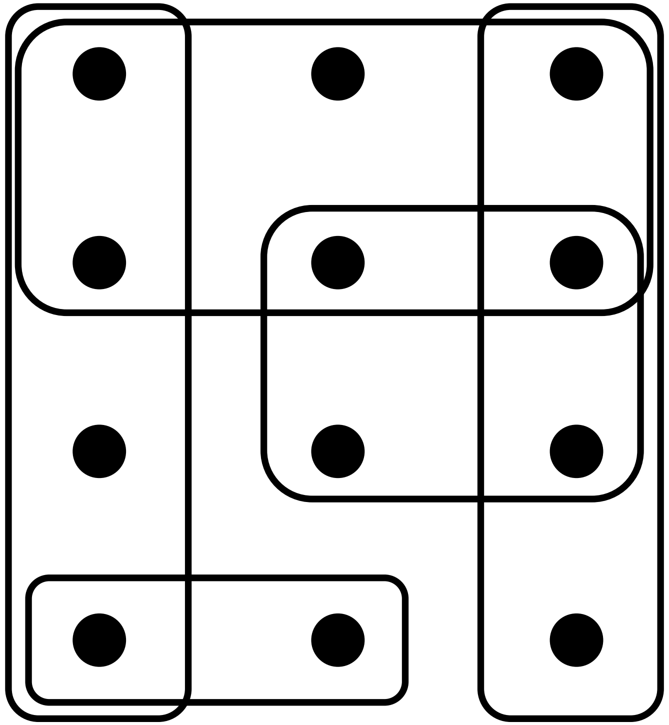
$$\{\{1, 7, 8\}, \{4, 8, 10\}\} \qquad \{\{1, 4, 7\}, \{7, 8\}\}$$

no 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:



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

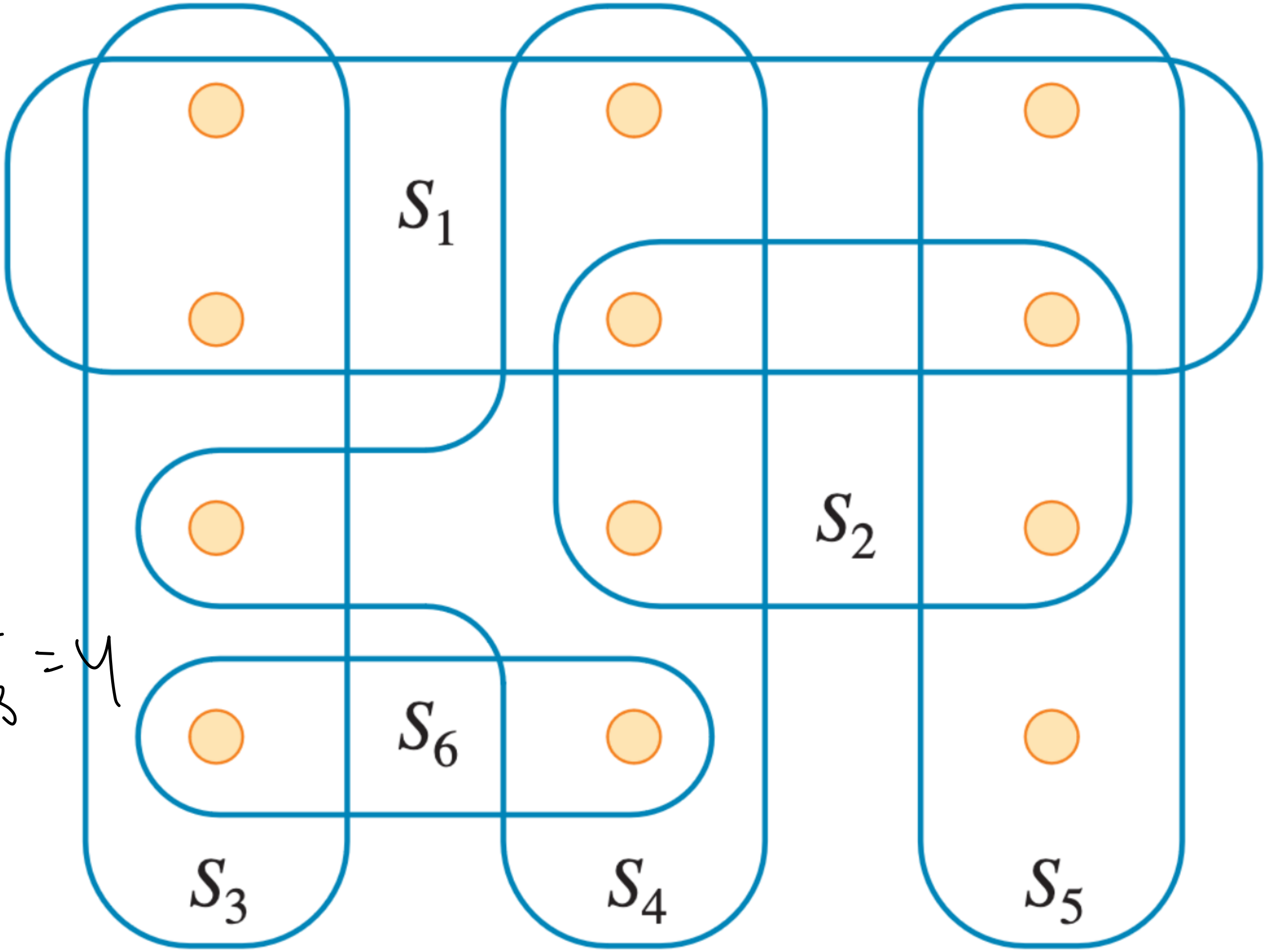
Set Cover: Given a set of elements (the universe), and sets containing those elements, find the smallest number of sets so that every element of the universe is included.

Greedy Algorithm:

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

Set Cover

select set w/ largest # excluded elements



$|S_1| = 6$

$6 \geq \frac{n}{OPT} = \frac{12}{13} = 4$

S_1, S_4, S_5, S_3

$n = 12$
$OPT = 3$

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?
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

ALG = ?

OPT = ?

Suppose the universe contains n elements.

$$\text{ALG} \leq \alpha \, \text{OPT}$$

Set Cover – Performance

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.

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

Set Cover – Performance

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

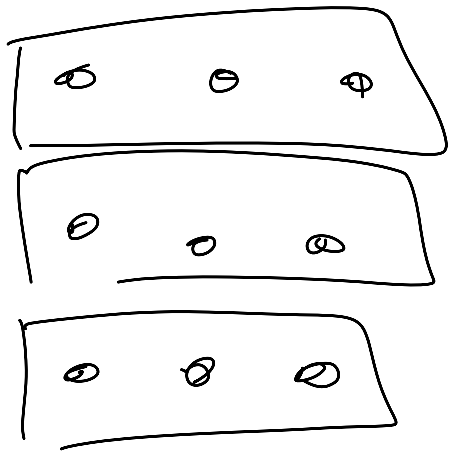
OPT = # sets in an optimal solution to cover all n elements.

Suppose the universe contains n elements.

The first set selected will have $\geq \frac{n}{\text{OPT}}$ elements because?

↳ in our algorithm aka the largest set

new # elements to cover $\leq n - \frac{n}{\text{OPT}}$



Set Cover – Performance

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 $\frac{n_{t-1}}{\text{OPT}}$ 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.

So, the universe is covered after at most $t = \text{OPT} \ln n$ iterations.

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

$\alpha = \ln(n)$

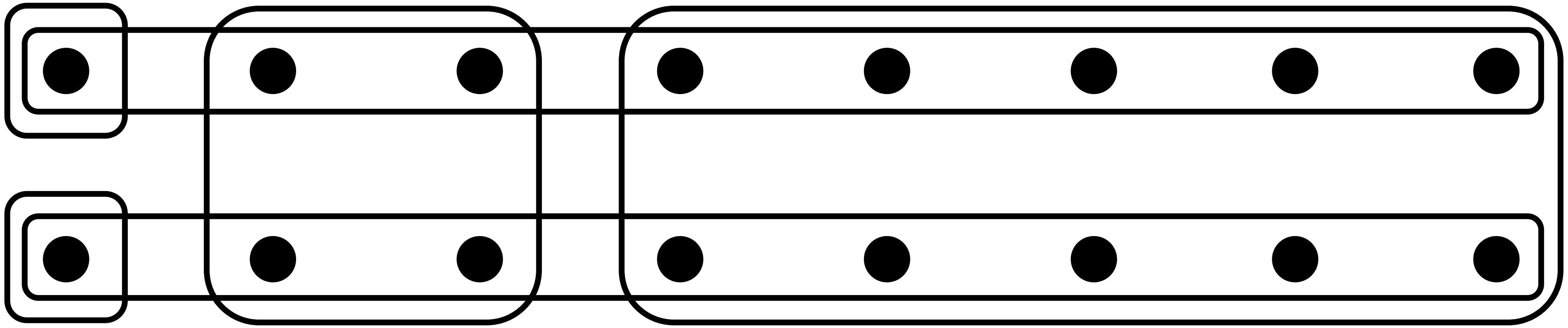
constant $\alpha = 2$

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 – Inapproximability

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

Set Cover – Inapproximability

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

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

Vertex Cover $\in$ APX
Set Cover $\notin$ APX