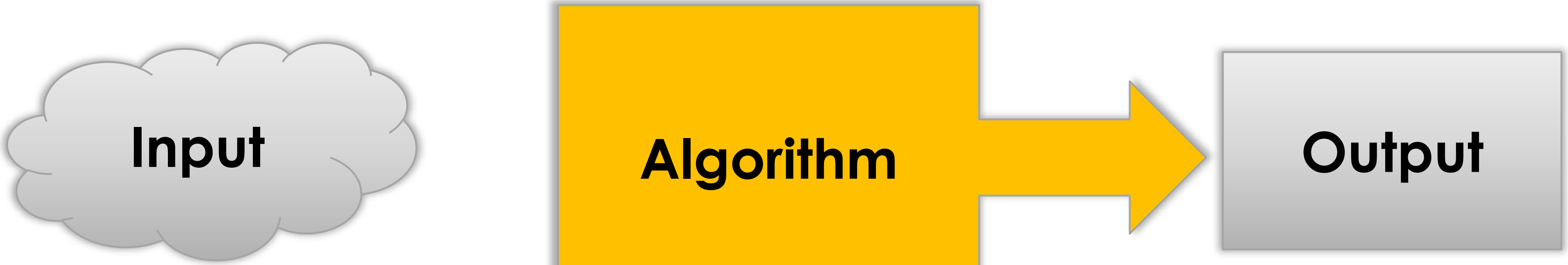
Algorithm performance

Best case, average case, worst case

By the end of this video you will be able to…

- Define worst case, average case, and best case performance
- Describe why each of these is used


Input
Algorithm
Output

```
public static boolean hasLetter (String word, char letter)
{
   for (int i = 0; i < word.length(); i++)
   {
      if (word.charAt(i) == letter)
      {
         return true;
      }
   }
   return false;
}
```

```
hasLetter("San Diego", 'S')
```

OR

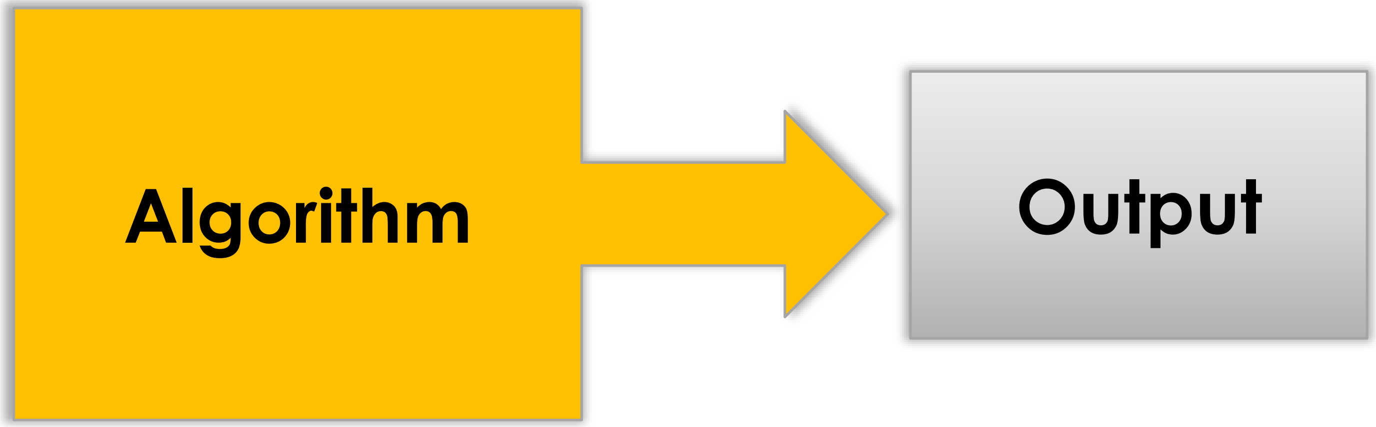
```
hasLetter("San Diego", 'i')
```

OR

```
hasLetter("San Diego", 'x')
```

OR

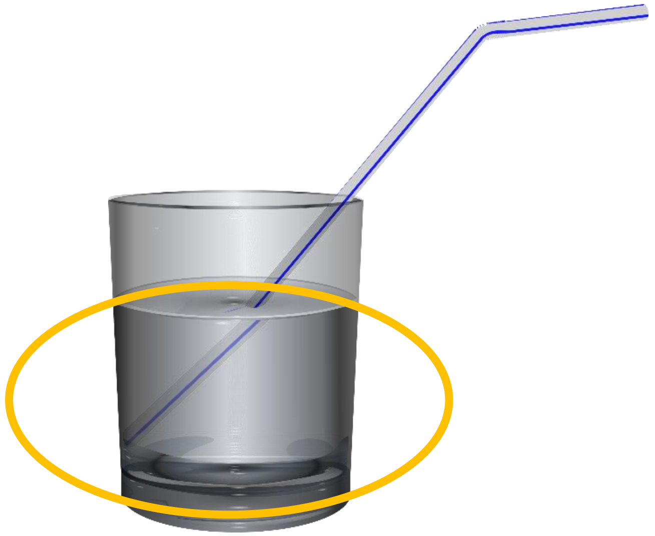
…



Best case

Best possible performance of algorithm for any input

(of fixed size n)



Best case

```
public static boolean hasLetter (String word, char letter)
{
   for (int i = 0; i < word.length(); i++)
   {
     if (word.charAt(i) == letter)
     {
        return true;
     }
   }
   return false;
}
```

Best case : word starts with letter

```
public static boolean hasLetter (String word, char letter)
{
  for (int i = 0; i < word.length(); i++)
  {
    if (word.charAt(i) == letter)
    {
       return true;
    }
  }
  return false;
}
```

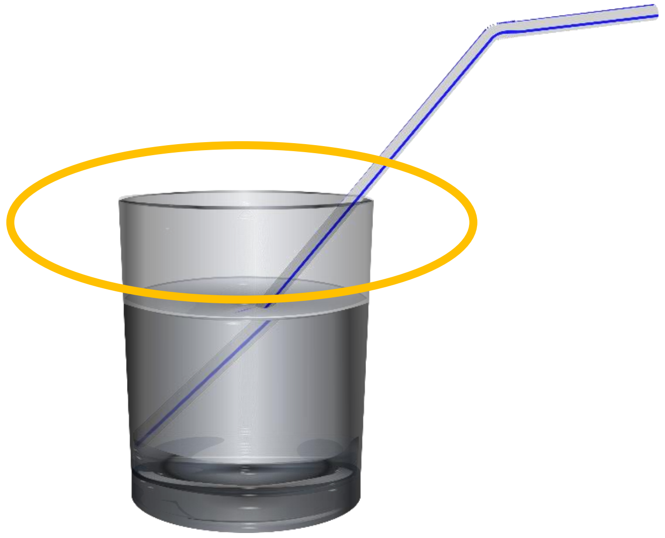
Best case : word starts with letter O(1)

```
public static boolean hasLetter (String word, char letter)
{
  for (int i = 0; i < word.length(); i++)
  {
    if (word.charAt(i) == letter)
    {
       return true;
    }
  }
  return false;
}
```

Worst case

Worst possible performance of algorithm for any input

(of fixed size n)



Worst case

```
public static boolean hasLetter (String word, char letter)
{
   for (int i = 0; i < word.length(); i++)
   {
      if (word.charAt(i) == letter)
      {
          return true;
      }
   }
   return false;
}
```

Worst case : letter at the end (or missing)

```
public static boolean hasLetter (String word, char letter)
{
   for (int i = 0; i < word.length(); i++)
   {
      if (word.charAt(i) == letter)
      {
          return true;
      }
   }
   return false;
}
```

Worst case : letter at the end (or missing) O(n)

```
public static boolean hasLetter (String word, char letter)
{
   for (int i = 0; i < word.length(); i++)
   {
      if (word.charAt(i) == letter)
      {
          return true;
      }
   }
   return false;
}
```

Best case $\leq$ Worst case

Average case

Performance of algorithm on average, consider all possible inputs of size n