Thinking Like an Engineer

Algorithms and Specificity

What is an algorithm?

It’s a step-by-step list of instructions explaining how to do something.

CARL SAGAN'S APPLE PIE

1 universe
1 9" pie shell
6 cups sliced apples
3/4 cup sugar
1/2 cup brown sugar
2 tbsp all-purpose flour
1/2 tsp cinnamon
1/8 tsp nutmeg
1/2 cup all-purpose flour
3 tbsp butter

Preparation time:
12-20 billion years

Servings:
8

Remember -
"If you want to make an apple pie from scratch, you must first create the universe."
–Carl

Preheat oven to 375 F. Make the universe as usual.

Place apples in a large bowl. In a smaller bowl, mix together sugar, 2 tbsp flour, cinnamon, and nutmeg. Sprinkle mixture over apples. Toss until evenly coated. Spoon mixture into pie shell.

In a small bowl mix together 1/2 cup flour and brown sugar. Add butter until mixture is crumbly. Sprinkle mixture over apples. Cover loosely with aluminum foil.

Bake in preheated oven for 25 minutes. Remove foil and bake another 30 minutes, or until golden brown.

So how do we create a procedure for consistently developing a great tasting recipe / algorithm?

We create a *Recipe* for an Algorithm

Recipe for an Algorithm

Recipe for an Algorithm

1. What are the guidelines of the problem?

What are the guidelines of the problem?

1. Clearly define the provided input(s).
2. Clearly define the expected output(s).
3. Clearly define any constraints.

Recipe for an Algorithm

1. What are the guidelines of the problem?
2. Analyze the problem space and consider possible solutions.

Analyze the problem space and consider possible solutions.

1. Is this problem similar to any that you've seen previously?
2. Are there any obvious edge-cases to consider?

Recipe for an Algorithm

1. What are the guidelines of the problem?
2. Analyze the problem space and consider possible solutions.
3. Create a step-by-step plan of action.

Create a step-by-step plan of action.

Example - Does the letter “G” exist in the provided string?

1. Assume that the string can be traversed in order.
2. Get a reference to the first item in the string.
3. Test - does that item equal “G”?
    a. If yes, return True
    b. if no, get a reference to the next item in the string and repeat Step 3.
4. If you get to the end of the list, return False

Recipe for an Algorithm

1. What are the guidelines of the problem?
2. Analyze the problem space and consider possible solutions.
3. Create a step-by-step plan of action.
4. If necessary, convert the plan into pseudocode.

Recipe for an Algorithm

1. What are the guidelines of the problem?
2. Analyze the problem space and consider possible solutions.
3. Create a step-by-step plan of action.
4. If necessary, convert the plan into pseudocode.
5. Translate the pseudocode into actual code.

```
var checkForG =  function(string){
   for (var i = 0; i < string.length; i++){
      if (string[i] === 'G'){
         return true;
      }
   }
   return false;
};
```

Test your Code