Algorithm Analysis Of Recurrence/Recursive Relations

In this part 2 of 5 we will understand and analyze complexity of a recursive relation using substitution method.

Algorithm Analysis And Design

Objective: in this blog post we will discuss how to analyze the complexity of a recurrence relation using substitution method. Steps and how to make a good guess related to the same.

Substitution method, as the name implies we substitute different values to find the complexity of the recursive function. Two important steps related are as follow:

1. Guess a particular solution in asymptotic form for the given recurrence relation.
2. Verify the guess using mathematical induction and find the value of constant C.

Let us understand substitution method using an example.

Consider the recurrence relation:

<script src="https://gist.github.com/NavneetPrakashSingh/70377dbe88618d68515f28af18f3385e.js"></script>

We wish to prove that T(n) = O(n2), then its sufficient to show that T(n) <= Cn2 . Hence the first part of substitution method is to make an assumption. So we make an assumption that T(a) <= Ca2 in order to prove T(n) = O(n2). Hence,

T(a) = Ca2 where 1 < a < n-1

The second step is to prove our assumption correct using mathematical induction. By substituting the induction hypothesis we get,

<script src="https://gist.github.com/NavneetPrakashSingh/c55cd02c03e16a96376c0a9096a9e0c1.js"></script>

We just proved using induction a recurrence relation using substitution method. We made an assumption and then proved our assumption is right using mathematical induction.

Trick for making a good guess

If a recurrence relation is of the form i.e.

T(n) = aT(n/b) + F(n)

Case 1 : If a = b = 2, then

T(n) = O(F(n)log n) is a good guess

Case 2 :If a = 1 and b = any value, then

T(n) = O(log n) is a good guess

Case 3: If a != 1 and b > a, then

T(n) = O[f(n) \* n] is a good guess

Now we know how to make a good guess in a substitution method.

That’s all folks. Time for a wrap-up. We have studied how to find complexity of a recurrence relation using substitution method and tried to understand what a good guess is. In the next tutorial we will discuss the next method i.e. Iterative method to find the complexity of a recurrence relation.