Binary Search: Divide-n-conquer

Problem Statement: Given a set of records R1, …, Rn with Keys K1, …, Kn such that K1<= K2<= …<=Kn and a particular key K, search whether the record corresponding to K exists.

BSearch:

I/O: A set of records R_1, …, R_n with Keys K_1, …, K_n such that $K_1 <= K_2 <= … <= K_n$ and a particular key K

O/P: Output “Yes” or “No”

Steps:

1. l<- 1, u<- n.
2. If l>u, Output “No” and return;
   mid <-floor((l+u)/2);
3. If $K=K_{mid}$, output “Yes” and return;
   Else if $K<K_{mid}$, go to step 4;
   Else go to Step 5;
4. u<- mid-1 and go to Step 2;
5. l<- mid+1 and go to Step 2;

Recurrence relation: T(n) =T(n/2) + c;

Task:

1. Write a program to implement BSearch. Find the time required for Best case, worst case and a random case.
2. Compare Sequential search algorithm (SSearch) with BSearch and find when SSearch will perform better than BSearch.
3. Take f(n)=c1.n and g(n) =c2.log(n). Plot the graph for the worst case of SSearch and BSearch along with these two functions. Find some constants for f(n) and g(n) such that the plot for SSearch is bounded by f(n) and BSearch is bounded by g(n).