Lab02-Divide and Conquer

Exercises for Algorithms by Xiaofeng Gao, 2018 Spring Semester.

∗ If there is any problem, please contact TA Jiaxi Liu.

∗

1. Assume array $A[1..n]$ is a *random* permutation of integers from 1 to n, please give the average case analysis of Algorithm 1. The exact expression for the average number (i.e., the expectation) of comparisons in Line 2 is required.

Algorithm 1: LinearSearch

Input: $A[1..n]$: a random permutation of integers between 1 and n.

```
1 for i ← 1 to n do
2     if A[i] == i then
3         return True;
4 return False;
```

Solution:

1. Let us assume the input is uniformly distributed. More Precisely to get the average case analysis of an Linear Search Algorithm normally you can simply go for $\frac{n+1}{2}$ due to the fact that an element x in an array of the size of n. That element can be at 1, or 2 or n location.

2. When we search we check each element in the array and compare it with x, and so when we search k^{th} element of the array we have already done k comparisons.

3. if it is at 1 then you find 1 comparison, if it's at 2, then you find 2 comparison.. if it is at n then you do n comparison in order to find it.

4. In order to find the average case of number of comparison we do $1 + 2 + ... + n = \frac{(n+1)n}{/}2$ and divide by n (size of the array) resulting in $\frac{n+1}{2}$

5. a more formal way to proof it would be as following: Let say X is a random variable for the number for comparisons. Then $p(X = x)$ will show the probability of the comparisons of X. Which we can note as $p(X = x) = \frac{1}{n}$.

6. $E[X] = \sum_{x=1}^{n} xp(x) = \sum_{x=1}^{n} \frac{x}{n} = \frac{1}{n} + \frac{2}{n} + ...\frac{n}{n} = \frac{n+1}{2}$

7. Yet there is one problem left. This doesn't include the $A[i] = i$ as stated in the assignment. Because i is random the chance of the value being the same number as the index is way smaller than $\frac{n+1}{2}$. Yet how to calculate this i'm not really sure. Would this mean $\frac{n+1}{2}$ divided by n?

2. Given an integer array $A[1..n]$ and two integers $lower \leq upper$, design an algorithm using divide-and-conquer method to count the number of ranges (i, j) $(1 \leq i \leq j \leq n)$ satisfying

$$lower \leq \sum_{k=i}^{j} A[k] \leq upper.$$

Example:

Given $A = [1, -1, 2]$, $lower = 1$, $upper = 2$, return 4.

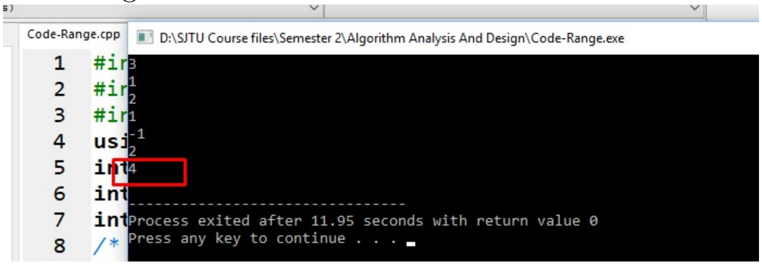
The resulting four ranges are $(1, 1)$, $(3, 3)$, $(2, 3)$ and $(1, 3)$.

1. Complete the implementation in the provided C/C++ source code (The source code *Code-Range.cpp* is attached on the course webpage).

2. Write a recurrence for the running time of the algorithm and solve it by recurrence tree (You can modify the figure source *Fig-RecurrenceTree.vsdx* to illustrate your derivation).

3. (Optional Sub-question with Bonus) Can we use the Master Theorem to solve the recurrence above? Please explain your answer.

Solution (a) The code is given in the file. (Code-Range.cpp) you should check it. The output result is given as follow.

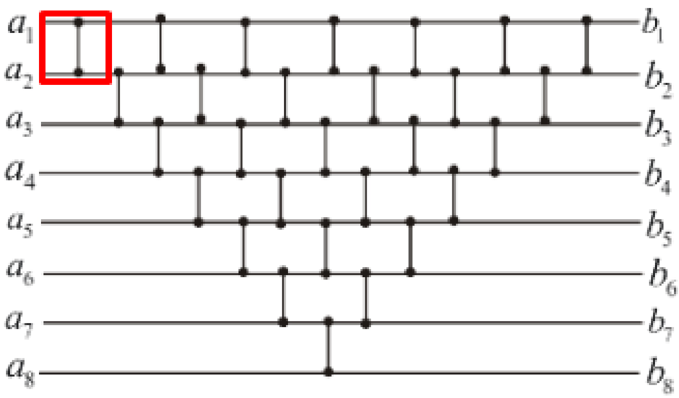


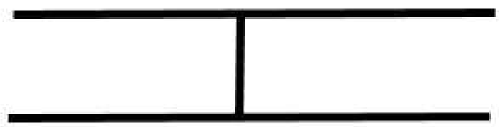
(b) As the equation states, The binary search is fesible for the divide and conquer method on the following algorithm, The recurrence for normal binary search is
$T(n) = T(\frac{n}{2}) + O(1)$.
This accounts for one comparison (on an element which partitions the n-element list of sorted keys into two $\frac{n}{2}$ elements set) and then a recursive call on the appropriate partition.

(c) For Master theorem:

Solution (3)

Proof: we firstly have to prove that if f is monotonically decreasing function, then a single comparator with inputs $f(x)$ and $f(y)$ produces outputs $f(min(x, y))$ and $f(max(x, y))$. We now use induction method to prove it.





$f(x)$ $\quad$ $max(f(x), f(y)) = f(max(x, y))$
$f(y)$ $\quad$ $max(f(x), f(y)) = f(max(x, y))$

f is monotonically decreasing
To prove above statement, assume a comparator whose input values are x and y. The above output is $max(x, y)$ is for upper output and $min(x, y)$ is lower output. Let we apply $f(x)$ and $f(y)$ to input of comparator as above the operation of comparator yield the value $min(f(x), f(y))$ on the above lower output and the value $max(x, y)$ on the upper output. Since f is monotonically decreasing.
$x \geq y \simeq f(x) \geq f(y)$
Consequently we have the identity
$min(f(x), f(y)) = f(min(x, y))$
$max(f(x), f(y)) = f(max(x, y))$
Thus the comparator produces the value of $max(x, y)$ and $min(x, y)$ when $f(x)$ and $f(y)$ are inputs.
So, this completes our proof.
we can prove the whole given figure by using this proof.