Data Structures and Algorithms II Assignment 1

Raul Rodriguez Castro

May 26, 2025

1 Median of Medians Proof

i) Let A be a set containing n elements. Then $|A| = n$.

ii) A is split into sub lists of 9 elements each.

iii) Let S be the set containing these sub lists such that $s_i \in S$, $0 \leq i \leq \lfloor n/9 \rfloor$ and $|s_i| = 9$.

iv) Let m be the median element of each s_i.

v) Let M be the set containing the medians of each sub list s_i, then $|M| = \lfloor n/9 \rfloor$.

vi) Let p be the median of M.

vii) Half of the elements in M are less than or equal to p meaning there are $n/18$ elements less than it.

viii) Since each median m has 4 elements $\leq m$ those four extra elements must be added to the previous calculation from vii) giving $(4n + n)/18$

ix) In the best case we must recurse the list with $5n/18$ elements, in the worst case we must recurse the other list containing at most $13n/18$.

$\therefore O(n) = T(13n/18) + T(n/9)$

2 Question 2

The following assumption has been made: P is always in the plain.

procedure BFPRT(A: Array$< T >$):
t $\leftarrow$ $\lfloor len(A/5) \rfloor$
//Partition A into t subarrays $S_0, S1, \ldots$
//Sort each sub array S_i
M $\leftarrow$ //Array of length t
for i $\leftarrow$ 0 to t - 1 do:
M[i] $\leftarrow$ $S_i[2]$ //store median in M
end
return Quickselect(M, $\lfloor t/2 \rfloor$ //split M until we have median of medians
end procedure

procedure Quickselect(A: Array $< T >$, k: int):
if len(A) = 1 then
return $A[0]$
end
p $\leftarrow$ BFPRT(A)
L $\leftarrow$ //elements to the left of pivot
E $\leftarrow$ //elements equal to pivot
G $\leftarrow$ //elements greater than pivot
if k ¡ len(L) then
return Quickselect(L, k)
end
if k $\geq$ len(L) + len(E):
return Quickselect(E, k - len(L) - len(E))
end
return E[k - len(L)]
end procedure

Point: (int, int)
S: Array$< Point >$
procedure FindClosestPoint(**P** : Point, **S**: Array$< Point >$, **k**: int): D $\leftarrow$ [] //distance vector
for i $\leftarrow$ 0 to len(S) - 1 do:
D[i] = $\sqrt{(P_0 - S[i]_0)^2 + (P_1 - S[i]_1)^2}$
end
kthSmallest $\leftarrow$ Quickselect(D, k)
return slice(D, 0, kThSmallest) // return the slice of the array grouping all elements $\leq k$

end procedure

We know BFPRT is $O(n)$, assuming the worst case $k = |S| - 1$, meaning all elements in D will be returned, we would be adding $2n$ more iterations. One to compute D and another to return the slice. $\therefore T(7n/10) + T(n/5) + cn + 2n$

3 Question 3

a.

Consider the best case where $A = [a_0, a_1, \ldots, a_n]$ is sorted. Then our comparison tree will have a height of $n-1$ as it is the number of comparisons that must be performed $\cdots < a_n, a_1 < \ldots, a_0 < a_1$.

let $p(n) : n - 1$

Base Case:

$1 - 1 = 0$ which holds because a 1 element list is sorted by definition, and no comparisons must be made. Therefore, our tree contains 0 nodes.

Induction Hypothesis:

Suppose $p(0) \ldots p(n)$ Hold.

Induction Step:

$n - 1$ holds by induction hypothesis for a list of size n. Then $n = n - 1 + 1$ comparisons are necessary for a list containing $n + 1$ elements.

Conclusion: By PMI $p(n)$ holds $\forall n \geq 1$.

b.

Consider the case where less than $n-1$ comparisons are performed in a list of n elements. Then there exists at least one element which has not been compared and thus, the correctness of the algorithm cannot be guaranteed.