Name: Alex Book
ID: 108073300

CSCI 3104, Algorithms **Profs. Hoenigman & Agrawal**
Problem Set 8a (14 points) **Fall 2019, CU-Boulder**

Advice 1: For every problem in this class, you must justify your answer: show how you arrived at it and why it is correct. If there are assumptions you need to make along the way, state those clearly.

Advice 2: Verbal reasoning is typically insufficient for full credit. Instead, write a logical argument, in the style of a mathematical proof.

Instructions for submitting your solution:

- The solutions **should be typed** and we cannot accept hand-written solutions. Here's a short intro to Latex.
- You should submit your work through **Gradescope** only.
- If you don't have an account on it, sign up for one using your CU email. You should have gotten an email to sign up. If your name based CU email doesn't work, try the identikey@colorado.edu version.
- Gradescope will only accept **.pdf** files (except for code files that should be submitted separately on Gradescope if a problem set has them) and **try to fit your work in the box provided**.
- You cannot submit a pdf which has less pages than what we provided you as Gradescope won't allow it.

Name: Alex Book
ID: 108073300

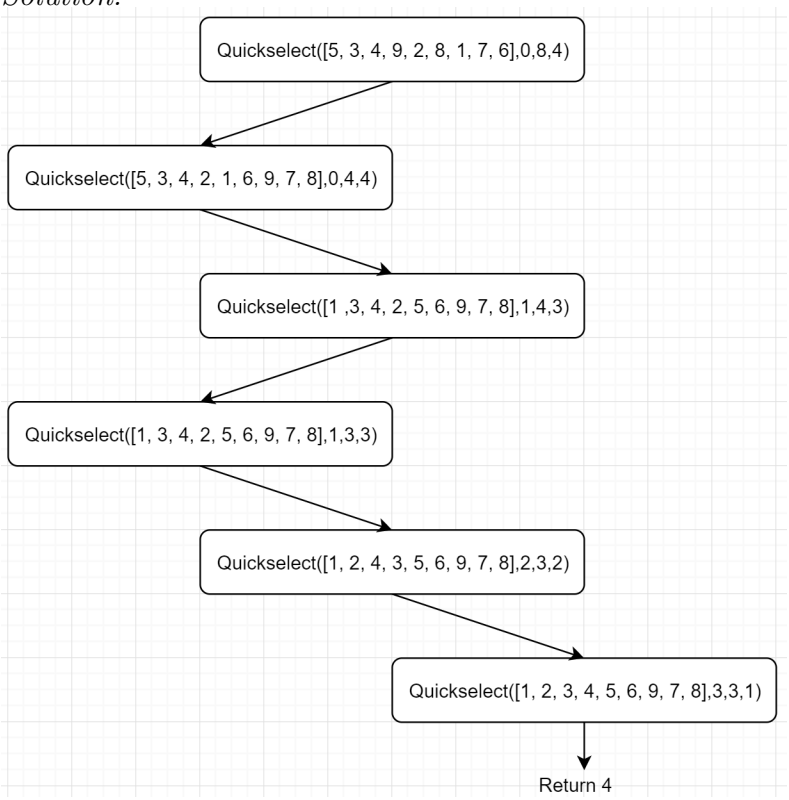
CSCI 3104, Algorithms **Profs. Hoenigman & Agrawal**
Problem Set 8a (14 points) **Fall 2019, CU-Boulder**

1. (2 pts) If the arrays, $A = [12, 14, 23, 34]$ and $B = [11, 13, 22, 35]$ are merged, list the indices in A and B that are compared to each other. For example, $A[0], B[0]$ means that $A[0]$ is compared to $B[0]$.

   Solution.
   A[0],B[0]
   A[0],B[1]
   A[1],B[1]
   A[1],B[2]
   A[2],B[2]
   A[2],B[3]
   A[3],B[3]

2. (3 pts) Illustrate how to apply the QuickSelect algorithm to find the $k = 4$th smallest element in the given array: `A = [5, 3, 4, 9, 2, 8, 1, 7, 6]` by showing the recursion call tree.

   Solution.



3. (1 pt) Explain in 2-3 sentences the purpose of the Median of Medians algorithm.

   Solution.
   The Median of Medians algorithm is used to select a 'good' pivot to be used to partition the given array (meaning it would allow the given array to be split into two groups of equal or nearly-equal size). It also does this selection in linear time, keeping the partition algorithm used in QuickSort/QuickSelect to an optimal running time, therefore keeping QuickSort/QuickSelect themselves at optimal running times.

4. (4 pts) Illustrate how to apply the Median of Medians algorithm (A Deterministic QuickSelect algorithm) to find the 4th largest element in the following array: `A = [6, 10, 80, 18, 20, 82, 33, 35, 0, 31, 99, 22, 56, 3, 32, 73, 85, 29, 60, 68, 99, 23, 57, 72, 25]`.

   Solution.
   MoMSelect([6, 10, 80, 18, 20, 82, 33, 35, 0, 31, 99, 22, 56, 3, 32, 73, 85, 29, 60, 68, 99, 23, 57, 72, 25],25,4)
   n>5, so split given array into sections of 5 elements each.
   $A_1 = [6, 10, 80, 18, 20]$
   $A_2 = [82, 33, 35, 0, 31]$
   $A_3 = [99, 22, 56, 3, 32]$
   $A_4 = [73, 85, 29, 60, 68]$
   $A_5 = [99, 23, 57, 72, 25]$
   Sort
   $A_1 = [6, 10, 18, 20, 80]$
   $A_2 = [0, 31, 33, 35, 82]$
   $A_3 = [3, 22, 32, 56, 99]$
   $A_4 = [29, 60, 68, 73, 85]$
   $A_5 = [23, 25, 57, 72, 99]$
   Medians = $[18, 33, 32, 68, 57]$
   Call MoMSelect on the array of medians. n$\leq$5, so sort the array.
   Medians = $[18, 32, 33, 57, 68]$
   Median of medians = 33
   Sort A using 33 as pivot
   $A = [6, 10, 18, 20, 33, 0, 31, 22, 3, 32, 29, 23, 25|80, 82, 35, 99, 56, 73, 85, 60, 68, 99, 57, 72]$
   Next iteration uses right half of this array as it has at least 4 elements, and we need the 4th largest element.
   Next iteration on next page

Solution.
MoMSelect([80,82,35,99,56,73,85,60,68,99,57,72],12,4)
n>5, so split given array into sections of 5 elements each (last section needn't have 5 elements).
$A_1 = [80, 82, 35, 99, 56]$
$A_2 = [73, 85, 60, 68, 99]$
$A_3 = [57, 72]$
Sort
$A_1 = [35, 56, 80, 82, 99]$
$A_2 = [60, 68, 73, 85, 99]$
$A_3 = [57, 72]$
Medians = $[80, 73, 64.5]$
Call MoMSelect on the array of medians. n$\leq$5, so sort the array.
Medians = $[64.5, 73, 80]$
Median of medians = 73
Sort A using 73 as pivot
$A = [35, 56, 73, 60, 68, 57, 72|80, 82, 99, 85, 99]$
Next iteration uses right half of this array as it has at least 4 elements, and we need the 4th largest element.
- - - - - - - - - -
MoMSelect([80,82,99,85,99],5,4)
n$\leq$5, so sort the array.
$A = [80, 82, 85, 99, 99]$
The 4th largest element in this subarray (and therefore in the initially given array) is the 4th-to-last element. Therefore we return 82 as the 4th largest element.

5. (4 pts) In Tuesday's lecture, we saw how the peaked array algorithm can find the maximum element in an array with one peak. For example, $A = [15, 16, 17, 14, 12]$ is a peaked array.

   (a) (2 pts) Explain how the peaked array algorithm works in sub-linear time? (You may use the recurrence relation to help with the explanation)

   Solution.
   The algorithm looks at the middle index. Based on that value and the values immediately next to it (before and after the middle), it either returns or recursively calls using half of the array. Therefore, it only evaluates a portion of the initially given array in order to find the peak, and works in sub-linear time.

   (b) (2 pts) Re-write the peaked array algorithm to find a single valley in an array, such as $A = [56, 43, 32, 21, 23, 25, 57]$. The valley would be 21.

   Solution.

```
findValley(A,p,r)
    mid=floor((p+r)/2)
    if A[mid] < A[mid+1] and A[mid] < A[mid-1]
        return A[mid]
    elif A[mid] > A[mid+1]
        return findValley(A,mid,r)
    return findValley(A,p,mid)
```