WIA2005 Algorithm Design & Analysis

Semester 2, 2016/17

Lab 3

PART 1

Implement the following algorithms:

- **Binary search**
- **Powering a number**
- **Fibonacci numbers**
- **Matrix multiplication**
- **Merge sort**
- **Quick sort**

PART 2

Finding the Matric Number

Suppose Mr Tommy was asked to key in student matric numbers into an unsorted array. It was told that the matric number of students were consecutives in nature from a minimum to a maximum number m and the total number of students were n. However, when Mr Tommy completed the process, he found out that one matric number was accidently left out that he has not keyed in. Assume $n= 2^k - 1$ and the matric numbers are integer numbers.

.

Implement the designed algorithm based on the above problem using the Divide and Conquer approach.

Solution: We can use SELECT to find the median element and check to see if it is in the array. If it is not, then it is the missing number. Otherwise, we PARTITION the array around the median element x into elements $\leq x$ and $> x$. If the first one has size less than $x + 1$, then we recurse on this subarray. Otherwise we recurse on the other subarray.

The procedure MISSINGINTEGER$(A, n, [i, j])$ takes as input an array A and a range $[i, j]$ in which the missing number lies.

```
MISSINGINTEGER(A, [i, j])
1  Determine median element x in range i ... j
2  Check to see if x is in A
3  PARTITION A into B, elements < x, and C, elements ≥ x
4  If SIZE(B) < x + 1
5     MISSINGINTEGER(B, [i, x])
6  Else MISSINGINTEGER(C, [x + 1, j])
```

The running time is $O(n)$ because the recurrence for this algorithm is $T(n) = T(n/2) + n$, which is $O(n)$ by the Master Method.

Common errors included using a randomized, instead of deterministic, partitioning scheme and using COUNTING SORT and then stepping through the array to find adjacent pairs that differ by two, which is not a Divide and Conquer approach.