|  |  |
| --- | --- |
| **Beirut Arab University**  **Faculty of Science**  **Computer Science Department** | **Course:**  CMPS441-Fundamentals of Algorithm  **Semester:** Fall 2024-2025  **Lab4** |

**Part I: Substitution Method and Master Theorem**

1. Using substitution method, determine a tight asymptotic lower bound for the following recurrences:
   1. T(n) = 4T(n/2) + n2.
   2. T(n) = 2T(n/2) + n.
2. For each of the following recurrences, give an expression for the runtime T(n) if the recurrence can be solved with the Master Theorem. Otherwise, indicate that the Master Theorem does not apply.

1. T (n) = 3T (n/2) + n2

2. T (n) = 4T (n/2) + n2

3. T (n) = 2nT (n/2) + nn

4. T (n) = 16T (n/4) + n

**Part II: Order Statistics: The k-SELECT problem**

Given an array and a number k where k is smaller than the size of the array, we need to find the k’th smallest element in the given array. It is given that all array elements are distinct.

**Examples:**

Input: arr[] = {7, 10, 4, 3, 20, 15}   
k = 3   
Output: 7

Input: arr[] = {7, 10, 4, 3, 20, 15}   
k = 4   
Output: 10

**Pseudocode:**

* ***Select****(A,k):*
  + ***If*** *len(A) <= 50:*
    - *A =* ***MergeSort****(A)*
    - ***Return*** *A[k-1]*
  + *p =* ***getPivot****(A)*
  + *L, pivotVal, R =* ***Partition****(A,p)*
  + ***if*** *len(L) == k-1:*
    - *return pivotVal*
  + ***Else if*** *len(L) > k-1:*
    - *return* ***Select****(L, k)*
  + ***Else if*** *len(L) < k-1:*
    - *return* ***Select****(R, k – len(L) – 1)*

***where:***

* ***getPivot****(A) returns some pivot for us:*
  + *Choose a random pivot; this takes time O(1)*
  + *Choose the pivot to be n/2; also takes time O(1)*
  + *Choose the pivot using merge sort algorithm, this finds the "best" pivot (that is, that splits the array in half); but in runs in time O(n log(n))*
* ***Partition****(A,p) splits up A into L, A[p], R.*