DESIGN AND ANALYSIS OF ALGORITHMS LABORATORY (MA39203)

Department of Mathematics
Indian Institute of Technology, Kharagpur
Assignment : 05
Date : 20/08/2025

Note: Use *iostream* header only for Question 1, 2 and 5.

1.(a) Create a class **MaxHeap** of your own that supports the following operations:

- **insert**(int *key*): Inserts a new element *key* into the heap,
- **getMax**(): Returns the maximum element of the heap,
- **extract_max**(): Removes and returns the maximum element from the heap,
- **max_heapify**(int i): Restores the max-heap property of the sub-tree rooted at index i,
- **build_heap**(int *arr*[ ], int n): Builds a heap from an input array *arr*[ ] of size n.
- **increaseKey**(int i, int k): Increases the value of the element at index i by k,
- **printHeap**(): Prints the contents of the heap in array form.

Use an array of size 1000 to manage the heap internally. You may also hard code inputs inside the main() function.

(b) Implement **HeapSort** to sort an array (in place) of integers in ascending order.

2. Implement **Radix Sort** algorithm using **Counting sort** as the auxiliary sort.

3. You are given two integers arrays **a**[ ] and **b**[ ] of same size n. For every index i, you have to modify a[i] by either adding b[i] to a[i] or subtracting b[i] from a[i], you are only allowed to do atmost K add operations. Find the largest possible sum of all the elements in **a**[ ] after doing the above operations.

Example: Input: $n = 5, K = 2$, a[ ]={1, 2, 3, 4, 5}, b[ ]={-1, 2, 3, 4, 5}   Output: 20
Explanation: You perform subtraction for the first 3 indices and addition for the last 2 indices to get the maximum possible sum $(1-(-1)) + (2-2) + (3-3) + (4+4) + (5+5) = 20$.

4. Given two integer arrays **a**[ ] and **b**[ ] of the same size, and an positive integer k, find the top k maximum sum combinations, where each combination is formed by adding one element from **a**[ ] and one from **b**[ ]. Return the k largest sums in descending order.

Example 1: Input: a[ ] = [3, 2], b[ ] = [1, 4], k = 2;   Output: [7, 6]
Explanation: Possible sums: $3 + 1 = 4, 3 + 4 = 7, 2 + 1 = 3, 2 + 4 = 6$, Top 2 sums are 7 and 6.

Example 2: Input: a[ ]=[1,4,2,3], b[ ]=[2,5,1,6], k=3; Output: [10, 9, 9]
Explanation: The top 3 maximum possible sums are : $4 + 6 = 10$, $3 + 6 = 9$, and $4 + 5 = 9$.

5. You are given an integer C. Among all possible pairs (A, B) satisfying the condition: $A \oplus B = C$ where $\oplus$ denotes the bit-wise XOR operation, find the pair (A, B) that maximizes the product $A \times B$. Additionally, the binary length of A and B must not exceed the binary length of C. The binary length of X is the number of bits in the binary representation of X.

Example: Input: $C = 6$;   Output: 5 3
There are 4 possible pairs: $5 \oplus 3 = 1 \oplus 7 = 4 \oplus 2 = 6 \oplus 0$; (5,3) have the maximum product.