# ALGORITHMS

SORTING (Set: 1)

## SOLUTION

**1.** The number of swaps needed to sort the numbers: 8, 22, 7, 9, 31, 19, 5, and 13 in ascending order using bubble sort is—

(a) 11 (b) 12
(c) 13 (d) 14

**Solution:** Option (d)

**Explanation:**

Net 14 comparisons are required:
8, 22, 7, 9, 31, 19, 5, 13 ❼ 8, 7, 9, 22, 19, 5, 31, 13

In this way next set of iteration goes on '⇧' denotes swap.
Net 14 comparisons are required.

**2.** Given 2 sorted list of size 'm' and 'n' respectively. The no. of comparisons needed in the worst case by the merge sort algorithm will be—

(a) m × n (b) maximum of m, n
(c) minimum of m, n (d) m + n -1

**Solution:** Option (d)

**Explanation:**

Each comparison gets one element in final sorted array. The last element needs no comparison.

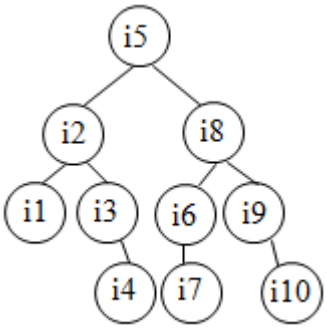
**3.** The average successful search time taken by binary search on a sorted array of 10 items is—

(a) 2.6 (b) 2.7
(c) 2.8 (d) 2.9

**Solution:** Option (d)

**Explanation:**

The 10 items i1, i2,…….i10 may be arranged in a BST as—




To get i5, 1 comparison
for i2 is 2
for i8 is 2
for i1 is 3 and so on.
(1+(2+2)+(3+3+3+3)+(4+4+4))/ 10= 2.9

**4.** The average successful search time for sequential search on 'n' items is—

(a) n/ 2
(b) (n-1)/ 2
(c) (n+1)/ 2
(d) log (n) +1

**Solution:** Option (c)

**Explanation:**

If the search key matches the very first item, with one comparison we can terminate. If it is second, 2 comparisons etc. So, average is→ (1+2+…..n)/n

. . $\frac{\square}{2} \times \frac{(\square + 1)}{\square} = \frac{\square + 1}{2}$

**5.** Sorting is useful for—

(a) report generation
(b) minimizing the storage needed
(c) making searching easier and efficient
(d) both (a) and (c)

**Solution:** Option (d)

**6.** Choose the correct statement—

(i) Internal sorting is used if the no. of items to be sorted is very large
(ii) External sorting is used if the no. of items to be sorted is very large
(iii) External sorting needs auxiliary storage
(iv) Internal sorting needs auxiliary storage

(a) (ii) and (iii)
(b) (i) and (iv)
(c) (ii) only
(d) (i) and (iii) only

**Solution:** Option (a)

**7.** A sorting technique that generates that records with the same primary key occurs in the same order in the sorted list as in the original unsorted list is said to be—

(a) stable
(b) consistent
(c) external
(d) linear

**Solution:** Option (a)

**8.** The way a card game player arranges his cards as he picks them up one by one, is an example of—

(a) bubble sort
(b) selection sort
(c) insertion sort
(d) merge sort

**Solution:** Option (c)

**9.** You want to check whether a given set of items is sorted. Which of the following sorting methods will be the most efficient if it is already in sorted order?

(a) Bubble sort
(b) Selection sort
(c) Insertion sort
(d) Merge sort

**Solution:** Option (c)

**Explanation :**

The complexity of insertion sort is O(n+ f(n)) where f(n) is the number of inversions. As f(n) is 0

for a sorted array hence we can use insertion sort technique to efficiently check if the array sorted.

**10.** The average no. of comparisons performed by the merge sort algorithm, in merging 2 sorted lists of length 2 is—

(a) 8/ 3
(b) 8/ 5
(c) 11/ 7
(d) 11/ 6

**Solution:** Option (a)

**Explanation:**

Merge sort combines 2 given sorted lists into 1 sorted list for this problem. Let the final sorted value is—a, b, c, d. The 2 lists (of length 2 each) should fall into one of the following 3 categories— (i) a, b and c, d
(ii) a, c and b, d
(iii) a, d and b, c.
The no. of comparisons needed in each case will be 2, 3, 3.
Therefore, Arrange no. of comparisons= (2+3+3)/ 3= 8/ 3

**11.** Which of the following sorting methods will be the best if no. of swapping done, is the only measure of efficiency?

(a) Bubble sort
(b) Selection sort
(c) Insertion sort
(d) Quick sort

**Solution:** Option (b)

**Explanation :**

For selection sort the number of swaps is O(n) . For bubble , insertion and quick sort , the number of swaps is O(n^2).

**12.** You are asked to sort 15 randomly generated numbers. One should prefer—

(a) Bubble sort
(b) Quick sort
(c) Merge sort
(d) Heap sort

**Solution:** Option (a)

**13.** As part of maintenance work, you are entrusted with the work of rearranging the library books in a shelf in proper order, at the end of each day. The ideal choice will be—

(a) Bubble sort
(b) Insertion sort
(c) Selection sort
(d) Heap sort

**Solution:** Option (b)

**14.** The max. no. of comparisons needed to sort 7 items using radix sort is (assume each item is a 4 digit decimal number)

(a) 280
(b) 40
(c) 47
(d) 38

**Solution:** Option (a)

**Explanation:**

The max. no. of comparison is no. of items × radix × no. of digits= $7 \times 10 \times 4 = 280$

**15.** Which of the following algorithm exhibits the unnatural behavior that, minimum no. of comparisons are needed if the list to be sorted is in the reverse sorted order and maximum no. of comparisons are needed if they are already in sorted order?

(a) Heap sort
(b) Radix sort
(c) Binary Insertion sort
(d) There can't be any such sorting method

**Solution:** Option (c)

**16.** Which of the following sorting algorithm has the worst time complexity of nlog(n) ?

(a) Heap sort
(b) Quick sort
(c) Insertion sort
(d) Selection sort

**Solution:** Option (a)

**17.** Which of the following sorting methods a given set of items that is already in sorted order?

(a) Heap sort
(b) Quick sort
(c) Insertion sort
(d) Selection sort

**Solution:** Option (c)

**18.** Merge sort uses—

(a) divide and conquer strategy
(b) backtracking approach
(c) heuristic search
(d) greedy approach

**Solution:** Option (a)

**19.** For merging 2 sorted lists of size m and n into a sorted list of size m + n, we require comparisons of—

(a) 0(m)
(b) 0(n)
(c) 0(m+n)
(d) 0(log(m) + log(n))

**Solution:** Option (c)

**20.** Which of the following design technique is used in the quick sort?

(a) Dynamic programming
(b) Backtracking
(c) Divide and conquer
(d) Greedy method

**Solution:** Option (c)