

1. Suppose we are comparing implementations of insertion sort and merge sort on the same machine. For inputs of size n , insertion sort runs in $8n^2$ steps, while mergesort runs in $64n\log_2 n$ steps. For which values of n does insertion sort beat mergesort? (1.2 – 2)

Answer : $8n^2 < 64n\log_2 n$

$\Rightarrow n^2 < 8n\log_2 n$

$\Rightarrow n < 8\log_2 n$

$\Rightarrow n < \log_2 n^8$

$\Rightarrow 2^n < n^8$

$n = 2$ to 43 Insertion sort will beat mergesort.

2. What is the smallest value of n such that an algorithm whose running time is $100n^2$ runs faster than an algorithm whose running time is 2^n on the same machine? (1.2 – 3)

Answer : $100n^2 < 2^n$

$n = 15$

3. Apply the insertion sort algorithm to sort the array $\langle 31, 41, 59, 26, 41, 58 \rangle$ (2.1 – 1)

Answer : $\langle 31, 41, 59, 26, 41, 58 \rangle$

$\Rightarrow \langle 31, 41, 59, 26, 41, 58 \rangle$

$\Rightarrow \langle 31, 41, 59, 26, 41, 58 \rangle$

$\Rightarrow \langle 26, 31, 41, 59, 41, 58 \rangle$

$\Rightarrow \langle 26, 31, 41, 41, 59, 58 \rangle$

$\Rightarrow \langle 26, 31, 41, 41, 58, 59 \rangle$

4. Rewrite the insertion sort pseudocode to sort into decreasing order instead of increasing order. (2.1 – 2)

Answer :

```
for j = 2 to A.length
    key = A[j]
    i = j - 1
    while i > 0 and A[i] < key
        A[i + 1] = A[i]
        i = i - 1
    end while
    A[i + 1] = key
end for
```

5. Consider the searching problem :

Input : A sequence of n numbers $A = \langle a_1, a_2, \dots, a_n \rangle$ and a value v

Output : An index i such that $v = A[i]$ or the special value NIL if v does not appear in A .

Write pseudocode for linear search, which scans through the sequence, looking for v . (2.1 – 3)

Answer :

```
i = NIL
for j = 1 to A.length
    if A[j] = v then
        i = j
        return i
    end if
end for
return i
```

6. Consider the problem of adding two n – bit binary integers, stored in two n – element arrays A and B . The sum of the two integers should be stored in binary form in an $(n + 1)$ – element array C . State the problem formally and write pseudocode for adding the two integers. (2.1 – 4)

Answer :

```
1111
1010
----
11001
carry = 0
for i = n to 1
    C[i + 1] = (A[i] + B[i] + carry)
    if A[i] + B[i] + carry ≥ 2 then
        carry = 1
    else
        carry = 0
    end if
end for
C[1] = carry
```