

**Test Bank for Data Abstraction and Problem Solving with C++ Walls and Mirrors**  
**7th Edition by Carrano Henry ISBN 0134463978 9780134463971**

Full link download:

Test Bank:

<https://testbankpack.com/p/test-bank-for-data-abstraction-and-problem-solving-with-c-walls-and-mirrors-7th-edition-by-carrano-henry-isbn-0134463978-9780134463971/>

Solution Manual:

<https://testbankpack.com/p/solution-manual-for-data-abstraction-and-problem-solving-with-c-walls-and-mirrors-7th-edition-by-carrano-henry-isbn-0134463978-9780134463971/>

## Chapter 2 Questions

### Multiple Choice Questions

1. In a recursive solution, the \_\_\_\_\_ terminates the recursive processing.
- a) local environment
  - b) pivot item
  - c) base case
  - d) recurrence relation

Answer: c.

2. A binary search uses a \_\_\_\_\_ strategy.
- a) divide-and-conquer
  - b) sequential
  - c) determine-the-pivot
  - d) smallest-to-largest

Answer: a.

3. A \_\_\_\_\_ is a mathematical formula that generates the terms in a sequence from previous terms.
- a) local environment
  - b) pivot item
  - c) base case
  - d) recurrence relation

Answer: d.

4. The factorial of  $n$  is equal to \_\_\_\_\_.
- a)  $n - 1$
  - b)  $n$  - factorial ( $n-1$ )
  - c) factorial ( $n-1$ )
  - d)  $n * \text{factorial}(n-1)$

Answer: d.

5. The base case for a recursive definition of the factorial of  $n$  is \_\_\_\_\_.
- a) factorial ( $-1$ )
  - b) factorial ( $0$ )
  - c) factorial ( $n$ )
  - d) factorial ( $n - 1$ )

Answer: b.

6. What is fundamentally wrong with computing the Fibonacci sequence recursively?

- a) it has two base cases
- b) each call to the function results in two recursive calls
- c) it computes the same values over and over
- d) nothing

Answer: c.

7. In the box trace, each box roughly corresponds to a(n)\_\_\_\_\_.

- a) recursive relation
- b) activation record
- c) base case
- d) pivot item

Answer: b.

8. In the box trace, each box contains all of the following EXCEPT\_\_\_\_\_.

- a) the values the function's arguments
- b) the function's local variables
- c) the function's execution time

## Chapter 2 Questions

- d) a placeholder for the value returned by each recursive call from the current box
- e) the value returned by the function itself

Answer: c.

9. In the box trace for a recursive function, a new box is created each time \_\_\_\_\_.

- a) the function is called
- b) the function returns a value
- c) an object is created
- d) an object is initialized

Answer: a.

10. What happens if a recursive function never reaches a base case?

- a) the function returns the correct value
- b) the function returns an incorrect value
- c) the function terminates immediately
- d) an infinite sequence of recursive calls occurs

Answer: d.

11. In a recursive method that writes a string of characters in reverse order, the base case is \_\_\_\_\_.

- a) a string with a length of 0
- b) a string whose length is a negative number
- c) a string with a length of 3
- d) a string that is a palindrome

Answer: a.

12. Which of the following is a precondition for a method that accepts a number  $n$  and computes the  $n^{\text{th}}$  Fibonacci number?

- a)  $n$  is a negative integer
- b)  $n$  is a positive integer
- c)  $n$  is greater than 1
- d)  $n$  is an even integer

Answer: b.

13. How many bases cases does a recursive binary search of a sorted array have?

- a) 0
- b) 1
- c) 2
- d) 3

Answer: c.

14. The number of ways to choose  $k$  out of  $n$  things is \_\_\_\_\_.

- a) the number of ways to choose  $k - 1$  out of  $n - 1$  things
- b) the number of ways to choose  $k$  out of  $n - 1$  things
- c) the sum of the number of ways to choose  $k - 1$  out of  $n - 1$  things and the number of ways to choose  $k$  out of  $n - 1$  things
- d) the product of the number of ways to choose  $k - 1$  out of  $n - 1$  things and the number of ways to choose  $k$  out of  $n - 1$  things

Answer: c.

15. When you solve a problem by solving two or more smaller problems, each of the smaller problems must be \_\_\_\_\_ the base case than the original problem.

- a) closer to
- b) farther to
- c) either closer to or the same “distance” from
- d) either farther to or the same “distance” from

Answer: a.

## Chapter 2 Questions

16. A recursive method that computes the number of groups of  $k$  out of  $n$  things has the precondition that \_\_\_\_\_.
- a)  $n$  is a positive number and  $k$  is a nonnegative number
  - b)  $n$  is a nonnegative number and  $k$  is a positive number
  - c)  $n$  and  $k$  are nonnegative numbers
  - d)  $n$  and  $k$  are positive numbers

Answer: c.

17. The midpoint of a sorted array has the index \_\_\_\_\_, where `first` is the index of the first item in the array, and `last` is the index of the last item in the array.

- a) `first / 2 + last / 2`
- b) `first / 2 - last / 2`
- c) `(first + last) / 2`
- d) `(first - last) / 2`

Answer: c.

18. If the value sought by a recursive binary search algorithm is in the array, which of the following is true?

- a) the algorithm makes the same comparisons as a sequential search
- b) the algorithm is successful without reaching a base case
- c) the algorithm searches the entire array
- d) the algorithm searches only the array half containing the value

Answer: d.

19. Which of the following is NOT a precondition for an array that is to be searched by a recursive binary search algorithm? (`first` is the index of the first item in the array, `last` is the index of the last item in the array, and `SIZE` is size of the array)

- a) `SIZE <= first`
- b) `0 <= first`
- c) `last <= SIZE - 1`
- d) `anArray[first] <= anArray[first + 1] <= ... <= anArray[last]`

Answer: a.

20. What does the following recursive algorithm display?

```
writeBack(in s:string)
  if (s is empty)
    return
  else
  {
    Write the first character of s
    writeBack(the string beginning at the second character of s)
  }
```

- a) nothing
- b) the first character of  $s$  a number of times equal to the length of  $s$
- c) the string  $s$
- d) the string  $s$  backward

Answer: c.

21. For an array containing 2, 3, 5, 6, 9, 13, 16, and 19, what value does a recursive binary search algorithm return when it searches for 6?

- a) 1
- b) 3
- c) 4
- d) none of the above

Answer: b.

## Chapter 2 Questions

22. For an array containing 2, 3, 5, 6, 9, 13, 16, and 19, what value does a recursive binary search algorithm return when it searches for 10?
- a) -1
  - b) 0
  - c) 1
  - d) 10

Answer: a.

23. In a sorted array having `SIZE` locations, the  $k^{\text{th}}$  smallest item is given by \_\_\_\_\_.
- a) `anArray[k-1]`
  - b) `anArray[k]`
  - c) `anArray[SIZE-k]`
  - d) `anArray[SIZE+k]`

Answer: a.

24. A recursive binary search algorithm always reduces the problem size by \_\_\_\_\_ at each recursive call.
- a) 1
  - b) 2
  - c) half
  - d) one-third

Answer: c.

25. A recursive solution that finds the factorial of  $n$  always reduces the problem size by \_\_\_\_\_ at each recursive call.
- a) 1
  - b) 2
  - c) half
  - d) one-third

Answer: a.

26. In the recursive solution to finding the  $k^{\text{th}}$  smallest item in an array, the problem size decreases by \_\_\_\_\_ at each recursive call.
- a) 1
  - b) at least 1
  - c) half
  - d) at least half

Answer: b.

27. In the recursive solution to the Towers of Hanoi problem, the number of disks to move \_\_\_\_\_ at each recursive call.
- a) decreases by 1
  - b) increases by 1
  - c) decreases by half
  - d) increases by half

Answer: a.

28. A recursive solution that finds the factorial of  $n$  generates \_\_\_\_\_ recursive calls.
- a)  $n - 1$
  - b)  $n$
  - c)  $n + 1$
  - d)  $n * 2$

Answer: b.

29. In the Fibonacci sequence, which of the following integers comes after the sequence 1, 1, 2, 3?
- a) 3
  - b) 4

## Chapter 2 Questions

c) 5

d) 6

Answer: c.

30. Which of the following is a base case for a recursive binary search algorithm?

(*first* is the index of the first item in the array, *last* is the index of the last item in the array, and *mid* is the midpoint of the array).

a) `last > first`

b) `first > last`

c) `0 <= first`

d) `last <= SIZE-1`

Answer: b.

## Chapter 2 Questions

### True/False Questions

1. A recursive solution solves a problem by solving a smaller instance of the same problem.  
Answer: True.
2. An iterative solution involves loops.  
Answer: True.
3. A binary search starts at the beginning of the collection of items.  
Answer: False.
4. An iterative method always calls itself.  
Answer: False.
5. In practice, recursion should be used even when a problem has a simple iterative solution. Answer: False.
6. When constructing a recursive solution, you should assume that a recursive call's postcondition is true if its precondition is true.  
Answer: True.
7. Every recursive method must have a base case. Answer: True.
8. A recursive solution can have more than one base case. Answer: True.
9. The binary search algorithm can be applied to an unsorted array.  
Answer: False.
10. The base case for a recursive solution to finding the  $k^{\text{th}}$  smallest item in an array cannot be predicted in advance. Answer: True.

## Chapter 2 Questions

### Short Answer Questions

1. What items does a sequential search examine when it is successful?

Answer: A sequential search starts at the beginning of the collection and looks at every item in the collection in order until the desired item is found.

2. What is a base case?

Answer: A base case is a special case of a recursive function whose solution you know. A base case enables the recursive processing to terminate.

3. What are the four questions that must be considered when constructing a recursive solution?

Answer: The four questions that must be considered when constructing a recursive solution are:

- How can you define the problem in terms of a smaller problem of the same type?
- How does each recursive call diminish the size of the problem?
- What instance of the problem can serve as the base case?
- As the problem size diminishes, will you reach this base case?

4. What is a recurrence relation?

Answer: A recurrence relation is a mathematical formula that generates the terms in a sequence from previous terms.

5. What is the box trace?

Answer: The box trace is a systematic way to trace the actions of a recursive method.

6. What is an activation record?

Answer: An activation record is a record that contains a method's local environment at the time of and as a result of the call to the method.

7. What elements are included in a method's local environment?

Answer: The local environment of a method includes the method's local variables, a copy of the actual value arguments, a return address in the calling routine, and the value of the method itself.

8. What are the two base cases for a recursive binary search

algorithm? Answer: The two base cases are:

- `first > last`
- `value == anArray[mid]`  
where `first` is the index of the first item in the array `anArray`, `last` is the index of the last item in `anArray`, and `mid` is the midpoint of `anArray`.

9. When is the base case `first > last` (where `first` is the index of the first item in the array and `last` is the index of the last item in the array) reached in a recursive binary search algorithm?

Answer: This base case is reached when the value sought is not in the original array.

10. When is the base case `value == anArray[mid]` (where `mid` is the midpoint of the array) reached in a recursive binary search algorithm?

Answer: This base case is reached when the value sought is in the original array.

11. What is a pivot item?

Answer: A pivot item is an element of a data collection that an algorithm uses to arrange the rest of the collection.

12. What is the base case for the recursive solution to the Towers of Hanoi problem?

Answer: The base case for the recursive solution to the Towers of Hanoi problem is a tower containing only one disk.

13. What are the two factors that contribute to the inefficiency of some recursive solutions?



## Chapter 2 Questions

Answer: The two factors are the overhead associated with method calls, and the inherent inefficiency of some recursive algorithms.

14. What is a tail-recursive function?

Answer: A tail-recursive function is a function whose last action is a recursive call.

15. Why do some compilers automatically replace tail recursion with iteration?

Answer: Some compilers automatically replace tail recursion with iteration because tail-recursive methods are often less efficient than their iterative counterparts, and the conversion of a tail-recursive method to an equivalent iterative method is rather mechanical.