Introduction to Algorithms and Data Structures 1st edition Cengage Test Bank

SKU: 9780357673560-TEST-BANK

Chapter 1 - Recursion

1. What is the base case for a recursive function to compute xn?
   a. $x = 0$
   b. $n = 0$
   c. $xn = 0$
   d. $x = 1$

ANSWER: b

FEEDBACK:
a. Incorrect. The first structure required in recursion is the base case, the case in which the function will no longer call itself. For a recursive function that computes xn, the base case is n = 0 (not $x = 0$). The base case is determined by the value of n, not x.
b. Correct. The first structure required in recursion is the base case, the case in which the function will no longer call itself. For a recursive function that computes xn, the base case is $n = 0$.
c. Incorrect. The first structure required in recursion is the base case, the case in which the function will no longer call itself. For a recursive function that computes xn, the base case is n = 0 (not $xn = 0$). The base case is determined by the value of n, not x.
d. Incorrect. The first structure required in recursion is the base case, the case in which the function will no longer call itself. For a recursive function that computes xn, the base case is n = 0 (not $x = 1$). The base case is determined by the value of n, not x.

POINTS: 1
DIFFICULTY: Introductory
REFERENCES: Introduction to Recursion
QUESTION TYPE: Multiple Choice
HAS VARIABLES: False
LEARNING OBJECTIVES: IADS.CENG.24.1.1.1 - Describe the use of recursion in defining the power of a number.
KEYWORDS: Bloom's: Understand
DATE CREATED: 7/31/2023 1:41 PM
DATE MODIFIED: 8/3/2023 1:40 PM

2. In tail recursion, which command in the function is the recursive call?
   a. the first one
   b. the last one
   c. the one with the base case condition
   d. the one that is the sum of two terms

ANSWER: b

FEEDBACK:
a. Incorrect. In tail recursion, the last (not the first) command in the function is the recursive call. The returned value in the recursive call case is only the recursive function.
b. Correct. In tail recursion, the last command in the function is the recursive call. The returned value in the recursive call case is only the recursive function.
c. Incorrect. In tail recursion, the last command in the function is the recursive call, not the one with the base case condition. The base case condition leads to a stopping point, not a recursive call.

d. Incorrect. In tail recursion, the last command in the function is the recursive call. The returned value in the recursive call case is only the recursive function. The function may or may not contain a command that is the sum of two terms.

POINTS: 1
DIFFICULTY: Introductory
REFERENCES: Introduction to Recursion
QUESTION TYPE: Multiple Choice
HAS VARIABLES: False
LEARNING OBJECTIVES: IADS.CENG.24.1.1.2 - Recall the conditions of well-defined recursive functions and procedures.
KEYWORDS: Bloom's: Remember
DATE CREATED: 7/31/2023 1:55 PM
DATE MODIFIED: 8/3/2023 1:40 PM

3. If recursion is not well-defined, it has no stopping case and will eventually result in a stack overflow error.

a. True
b. False

ANSWER: True
FEEDBACK:
Correct Well-defined recursion is a term used to describe a recursive algorithm that reaches an end state in a finite time and returns the same value every time it is used. If recursion is not well-defined, it has no stopping case and will eventually result in a stack overflow error.
Incorrect Well-defined recursion is a term used to describe a recursive algorithm that reaches an end state in a finite time and returns the same value every time it is used. If recursion is not well-defined, it has no stopping case and will eventually result in a stack overflow error.

POINTS: 1
DIFFICULTY: Intermediate
REFERENCES: Introduction to Recursion
QUESTION TYPE: True / False
HAS VARIABLES: False
LEARNING OBJECTIVES: IADS.CENG.24.1.1.3 - Associate the condition of well-defined recursion with a stack overflow error.
KEYWORDS: Bloom's: Understand
DATE CREATED: 7/31/2023 1:58 PM
DATE MODIFIED: 8/3/2023 1:40 PM

4. What is the base case for a recursive function to compute n! (factorial of n)?

a. $n^2 = 1$
b. $n = 2$
c. $n = 1$
d. $n = 0$

ANSWER: d
FEEDBACK:
a. Incorrect. The base case for a recursive function to compute n! (factorial of n) is n

= 0 (not n^2 = 1). The factorial of a non-negative integer n, usually written as "n!", is the product of all positive integers less than n. Calculating the factorial of an integer does not involve squaring it.

b. Incorrect. The base case for a recursive function to compute n! (factorial of n) is n = 0 (not n = 2). If n = 2, calculating the factorial of n will require recursion: `n * factorial(n - 1)`.

c. Incorrect. The base case for a recursive function to compute n! (factorial of n) is n = 0 (not n = 1). If n = 1, calculating the factorial of n will require recursion: `n * factorial(n - 1)`.

d. Correct. The base case for a recursive function to compute n! (factorial of n) is n = 0. The factorial of a non-negative integer n, usually written as "n!", is the product of all positive integers less than n.

POINTS:	1
DIFFICULTY:	Introductory
REFERENCES:	Introduction to Recursion
QUESTION TYPE:	Multiple Choice
HAS VARIABLES:	False
LEARNING OBJECTIVES:	IADS.CENG.24.1.1.4 - Describe the relationship between recursion and the divide-and-conquer strategy of problem-solving.
KEYWORDS:	Bloom's: Understand
DATE CREATED:	7/31/2023 1:59 PM
DATE MODIFIED:	8/3/2023 1:40 PM

5. How many recursive calls are made in a recursive algorithm computing the Fibonacci sequence for n = 3?

a. 3
b. 4
c. 5
d. 6

ANSWER: b

FEEDBACK:

a. Incorrect. It requires 4 recursive calls, not 3. For a recursive function fib, fib(3) calls fib(2) and fib(1). fib(2) calls fib(1) and fib(0). The base cases are n = 0 and n = 1, and making two recursive calls to the function is necessary to compute a single number.

b. Correct. It requires 4 recursive calls. For a recursive function fib, fib(3) calls fib(2) and fib(1). fib(2) calls fib(1) and fib(0). The base cases are n = 0 and n = 1, and making two recursive calls to the function is necessary to compute a single number.

c. Incorrect. It requires 4 recursive calls, not 5. For a recursive function fib, fib(3) calls fib(2) and fib(1). fib(2) calls fib(1) and fib(0). The base cases are n = 0 and n = 1, and making two recursive calls to the function is necessary to compute a single number.

d. Incorrect. It requires 4 recursive calls, not 6. For a recursive function fib, fib(3) calls fib(2) and fib(1). fib(2) calls fib(1) and fib(0). The base cases are n = 0 and n = 1, and making two recursive calls to the function is necessary to compute a single number.

POINTS:	1
DIFFICULTY:	Introductory
REFERENCES:	Introduction to Recursion
QUESTION TYPE:	Multiple Choice
HAS VARIABLES:	False
LEARNING OBJECTIVES:	IADS.CENG.24.1.1.5 - Describe the use of tail recursion in recursive functions and procedures.
KEYWORDS:	Bloom's: Remember
DATE CREATED:	7/31/2023 2:07 PM
DATE MODIFIED:	8/3/2023 1:40 PM

6. Which term describes a recursive algorithm that reaches an end state in a finite time and returns the same value every time it is used?

a. tail recursion

b. infinite recursion

c. well-defined recursion

d. stack overflow recursion

ANSWER: c

FEEDBACK:

a. Incorrect. Well-defined recursion is a term used to describe a recursive algorithm that reaches an end state in a finite time and returns the same value every time it is used. Tail recursion is a form of recursion where the last command in the function is the recursive call.

b. Incorrect. Well-defined recursion is a term used to describe a recursive algorithm that reaches an end state in a finite time and returns the same value every time it is used. Infinite recursion occurs when the function doesn’t reach a point where it stops because it is *not* well defined.

c. Correct. Well-defined recursion is a term used to describe a recursive algorithm that reaches an end state in a finite time and returns the same value every time it is used.

d. Incorrect. Well-defined recursion is a term used to describe a recursive algorithm that reaches an end state in a finite time and returns the same value every time it is used. A stack overflow error occurs when the stack doesn’t have space to store necessary data. Because each call of a function uses stack memory, recursive functions can trigger stack overflow errors when they are poorly defined.

POINTS:	1
DIFFICULTY:	Introductory
REFERENCES:	Examples of Recursive Methods
QUESTION TYPE:	Multiple Choice
HAS VARIABLES:	False
LEARNING OBJECTIVES:	IADS.CENG.24.1.2.1 - Associate the recursive definition of factorials with its non-recursive definition.
KEYWORDS:	Bloom's: Remember
DATE CREATED:	7/31/2023 2:09 PM
DATE MODIFIED:	8/3/2023 1:40 PM

7. The use of recursion enables you to create solutions using the divide-and-conquer strategy.

a. True
b. False

ANSWER: True

FEEDBACK:

Correct Recursion is a technique to solve problems where the case at hand is divided into smaller subtasks similar to the original one, and is useful when solving a problem or executing a task that you can split into smaller tasks. Thus, the use of recursion enables you to create solutions using the divide-and-conquer strategy.

Incorrect Recursion is a technique to solve problems where the case at hand is divided into smaller subtasks similar to the original one, and is useful when solving a problem or executing a task that you can split into smaller tasks. Thus, the use of recursion enables you to create solutions using the divide-and-conquer strategy.

POINTS: 1
DIFFICULTY: Advanced
REFERENCES: Examples of Recursive Methods
QUESTION TYPE: True / False
HAS VARIABLES: False
LEARNING OBJECTIVES: IADS.CENG.24.1.2.2 - Calculate the Fibonacci sequence using its recursive definition.
KEYWORDS: Bloom's: Apply
DATE CREATED: 7/31/2023 2:11 PM
DATE MODIFIED: 8/3/2023 1:40 PM

8. If a function `func1` calls a function `func2`, which calls a function `func3`, which calls the `func1`, what kind of recursion is this?

a. infinite recursion
b. tail recursion
c. direct recursion
d. indirect recursion

ANSWER: d

FEEDBACK:

a. Incorrect. Indirect recursion is when a function calls to itself but in an indirect manner, such as in this example. Infinite recursion occurs when the function doesn’t reach a point when it stops.
b. Incorrect. Indirect recursion is when a function calls to itself but in an indirect manner, such as in this example. Tail recursion is a form of recursion where the last command in the function is the recursive call.
c. Incorrect. Indirect recursion is when a function calls to itself but in an indirect manner, such as in this example. Direct recursion is when the function calls itself directly.
d. Correct. Indirect recursion is when a function calls to itself but in an indirect manner, such as in this example.

POINTS: 1
DIFFICULTY: Intermediate
REFERENCES: Direct and Indirect Recursion
QUESTION TYPE: Multiple Choice
HAS VARIABLES: False
LEARNING OBJECTIVES: IADS.CENG.24.1.3.1 - Describe the use of direct and indirect recursion for the computation of squares of a number.

KEYWORDS:	Bloom's: Apply
DATE CREATED:	7/31/2023 2:11 PM
DATE MODIFIED:	8/3/2023 1:40 PM

9. When using indirect recursion, you must be sure that in each cycle of going from one function to another, the parameters are increasing; otherwise, you may potentially enter an infinite recursion state.

a. True

b. False

ANSWER:	False
FEEDBACK:	*Correct* When using indirect recursion, you must be sure that in each cycle of going from one function to another, the parameters are decreasing, not increasing. If they are not decreasing, you have a circular definition, you are not using indirect recursion, and you may potentially enter an infinite recursion state. *Incorrect* When using indirect recursion, you must be sure that in each cycle of going from one function to another, the parameters are decreasing, not increasing. If they are not decreasing, you have a circular definition, you are not using indirect recursion, and you may potentially enter an infinite recursion state.
POINTS:	1
DIFFICULTY:	Intermediate
REFERENCES:	Direct and Indirect Recursion
QUESTION TYPE:	True / False
HAS VARIABLES:	False
LEARNING OBJECTIVES:	IADS.CENG.24.1.3.2 - Identify one consequence of circular definitions in code.
KEYWORDS:	Bloom's: Understand
DATE CREATED:	7/31/2023 2:13 PM
DATE MODIFIED:	8/3/2023 1:40 PM

10. It is easy to solve the Tower of Hanoi puzzle, and to describe the solution, when you do so with just two or three disks.

a. True

b. False

ANSWER:	True
FEEDBACK:	*Correct* The Tower of Hanoi puzzle is easy to solve and to describe the solution for when you have only two or three disks. *Incorrect* The Tower of Hanoi puzzle is easy to solve and to describe the solution for when you have only two or three disks.
POINTS:	1
DIFFICULTY:	Introductory
REFERENCES:	The Tower of Hanoi
QUESTION TYPE:	True / False
HAS VARIABLES:	False
LEARNING OBJECTIVES:	IADS.CENG.24.1.4.1 - Describe the problem of the Tower of Hanoi, its restriction, and the desired solution.
KEYWORDS:	Bloom's: Understand
DATE CREATED:	7/31/2023 2:14 PM
DATE MODIFIED:	8/3/2023 1:40 PM

11. In a recursive solution to the Tower of Hanoi problem for two disks, two disk movements are required.
    a. True
    b. False

ANSWER: False

FEEDBACK:
Correct In a recursive solution to the Tower of Hanoi problem for two disks, three (not two) disk movements are required. One to move the top disk from A to B, one to move the second disk from A to C, and one to move the disk from B to C.
Incorrect In a recursive solution to the Tower of Hanoi problem for two disks, three (not two) disk movements are required. One to move the top disk from A to B, one to move the second disk from A to C, and one to move the disk from B to C.

POINTS: 1
DIFFICULTY: Advanced
REFERENCES: The Tower of Hanoi
QUESTION TYPE: True / False
HAS VARIABLES: False
LEARNING OBJECTIVES: IADS.CENG.24.1.4.2 - Relate the solution algorithm presented with the concept of recursion.
KEYWORDS: Bloom's: Understand
DATE CREATED: 7/31/2023 2:14 PM
DATE MODIFIED: 8/3/2023 1:40 PM

12. Consider the problem finding a subset of {1, 2, 3, 4, 5, 6, 7, 8} in which the sum of the elements in the subset add up to 10. In a brute-force approach to this problem, what is the maximum possible number of subsets you would need to test to see if they add up to 10?
    a. 8
    b. 10
    c. 256
    d. 1024

ANSWER: c

FEEDBACK:
a. Incorrect. To find the desired solution it might be necessary to test all possible subsets, that is, $2n$ where n is the size of the original set. In this case, the original set has 8 elements and $2^8 = 256$, so there are 256 possible subsets to test, not 8. The number of possible subsets depends on the number of elements in the original target set (8), but calculating the number of possible subsets requires raising 2 to the power of this number.
b. Incorrect. To find the desired solution it might be necessary to test all possible subsets, that is, $2n$ where n is the size of the original set. In this case, the original set has 8 elements and $2^8 = 256$, so there are 256 possible subsets to test, not 10. The number of possible subsets does not depend upon the target sum (10), just on the number of elements in the original set.
c. Correct. To find the desired solution it might be necessary to test all possible subsets, that is, $2n$ where n is the size of the original set. In this case, the original set has 8 elements and $2^8= 256$, so there are 256 possible subsets to test.
d. Incorrect. To find the desired solution it might be necessary to test all possible subsets, that is, $2n$ where n is the size of the original set. In this case, the original

set has 8 elements and $2^8 = 256$, so there are 256 possible subsets to test, not 1024. The number of possible subsets is not 2 to the power of the target sum (10), but 2 to the power 8, which is the number of elements in the original set.

POINTS: 1
DIFFICULTY: Advanced
REFERENCES: Backtracking: Finding all Subsets
QUESTION TYPE: Multiple Choice
HAS VARIABLES: False
LEARNING OBJECTIVES: IADS.CENG.24.1.5.1 - State the difficulties of a pure brute force approach to the problem of finding all subsets of a given set.
KEYWORDS: Bloom's: Apply
DATE CREATED: 7/31/2023 2:15 PM
DATE MODIFIED: 8/3/2023 1:40 PM

13. What does backtracking use to determine if the search must continue or not?
    a. a bounding function/condition
    b. a recursive case
    c. the base case/condition
    d. tail recursion

ANSWER: a
FEEDBACK:

a. Correct. Backtracking uses a bounding function/condition to determine if the search must continue or not. A bounding condition tells you if you must keep following a certain path of options or if the path you are following is no longer factual for a solution.

b. Incorrect. Backtracking uses a bounding function/condition (not a recursive case) to determine if the search must continue or not. A bounding condition tells you if you must keep following a certain path of options or if the path you are following is no longer factual for a solution. A recursive function calls itself when it encounters a recursive case and stops calling itself when it arrives at a base case.

c. Incorrect. Backtracking uses a bounding function/condition (not a base case/condition) to determine if the search must continue or not. A bounding condition tells you if you must keep following a certain path of options or if the path you are following is no longer factual for a solution. A recursive function calls itself when it encounters a recursive case and stops calling itself when it arrives at a base case or condition.

d. Incorrect. Backtracking uses a bounding function/condition (not tail recursion) to determine if the search must continue or not. A bounding condition tells you if you must keep following a certain path of options or if the path you are following is no longer factual for a solution. In tail recursion, the last command in the function is the recursive call.

POINTS: 1
DIFFICULTY: Intermediate
REFERENCES: Backtracking: Finding all Subsets
QUESTION TYPE: Multiple Choice
HAS VARIABLES: False
LEARNING OBJECTIVES: IADS.CENG.24.1.5.2 - Apply the described algorithm to systematically enumerate all

	subsets of a given set.
KEYWORDS:	Bloom's: Remember
DATE CREATED:	7/31/2023 2:22 PM
DATE MODIFIED:	8/3/2023 1:40 PM

Name:____ Class:_________________________ Date:_____________

Chapter 1 - Recursion

1. What is the base case for a recursive function to compute xn?
   a. $x = 0$
   b. $n = 0$
   c. $xn = 0$
   d. $x = 1$

ANSWER: b

2. In tail recursion, which command in the function is the recursive call?
   a. the first one
   b. the last one
   c. the one with the base case condition
   d. the one that is the sum of two terms

ANSWER: b

3. If recursion is not well-defined, it has no stopping case and will eventually result in a stack overflow error.
   a. True
   b. False

ANSWER: True

4. What is the base case for a recursive function to compute n! (factorial of n)?
   a. $n^2 = 1$
   b. $n = 2$
   c. $n = 1$
   d. $n = 0$

ANSWER: d

5. How many recursive calls are made in a recursive algorithm computing the Fibonacci sequence for $n = 3$?
   a. 3
   b. 4
   c. 5
   d. 6

ANSWER: b

6. Which term describes a recursive algorithm that reaches an end state in a finite time and returns the same value every time it is used?
   a. tail recursion
   b. infinite recursion
   c. well-defined recursion
   d. stack overflow recursion

ANSWER: c

7. The use of recursion enables you to create solutions using the divide-and-conquer strategy.
   a. True
   b. False

ANSWER: True

8. If a function `func1` calls a function `func2`, which calls a function `func3`, which calls the `func1`, what kind of recursion is this?
   a. infinite recursion
   b. tail recursion
   c. direct recursion
   d. indirect recursion

ANSWER: d

9. When using indirect recursion, you must be sure that in each cycle of going from one function to another, the parameters are increasing; otherwise, you may potentially enter an infinite recursion state.
   a. True
   b. False

ANSWER: False

10. It is easy to solve the Tower of Hanoi puzzle, and to describe the solution, when you do so with just two or three disks.
   a. True
   b. False

ANSWER: True

11. In a recursive solution to the Tower of Hanoi problem for two disks, two disk movements are required.
   a. True
   b. False

ANSWER: False

12. Consider the problem finding a subset of {1, 2, 3, 4, 5, 6, 7, 8} in which the sum of the elements in the subset add up to 10. In a brute-force approach to this problem, what is the maximum possible number of subsets you would need to test to see if they add up to 10?
   a. 8
   b. 10
   c. 256
   d. 1024

ANSWER: c

13. What does backtracking use to determine if the search must continue or not?
   a. a bounding function/condition
   b. a recursive case
   c. the base case/condition
   d. tail recursion

ANSWER: a