Learning Activity: the C Language, Pointers, Malloc and Free

Due Sep 23 at 12:30am **Points** 10 **Questions** 10 **Available** after Sep 18 at 2:30am **Time Limit** None
Allowed Attempts Unlimited

Take the Quiz Again

Attempt History

	Attempt	Time	Score
KEPT	Attempt 8	less than 1 minute	10 out of 10
LATEST	Attempt 8	less than 1 minute	10 out of 10
	Attempt 7	less than 1 minute	9 out of 10
	Attempt 6	3 minutes	9 out of 10
	Attempt 5	2 minutes	8.83 out of 10
	Attempt 4	3 minutes	7.42 out of 10
	Attempt 3	28 minutes	4.75 out of 10
	Attempt 2	31 minutes	4.75 out of 10
	Attempt 1	3 minutes	5.08 out of 10

Score for this attempt: **10** out of 10

Submitted Sep 21 at 11:22pm

This attempt took less than 1 minute.

Question 1

1 / 1 pts

What happens if we forget to reclaim our memory with the "free" function?

Correct!

● A memory leak!

It is possible nothing happens if your system has lots of memory. However, if you run out of memory, your system will crash and need to restart to reclaim resources.

○ Nothing

Question 2

1 / 1 pts

I can always dereference a pointer.

○ True

Correct!

● False

Question 3

1 / 1 pts

Which of the following apply to a pointer?

Correct! ☑ A pointer can store the address of custom data types (structs).

Correct! ☑ A pointer is a data type

□ A pointer can refer to any type.

□ A pointer stores its data in octodecimal

Correct! ☑ A pointer stores an address

□ A pointer can't be accessed in C

Question 4

1 / 1 pts

In C, do we need to define "data types" for variables?

Correct! ● Yes, unlike languages like Python, when we declare something at "int" it always holds integer values

○ No, C is like Python where the type of the variable is inferred from its initial value

Question 5

1 / 1 pts

Which of the following are types provided by the C programming language?

Correct! ☑ double

Correct! ☑ integer

Correct! ☑ char

Correct! ☑ float

□ boolean

Question 6

1 / 1 pts

Is there a difference between the stack and the heap?

○ Yes, stack memory is automatically allocated and cleared when a function is called. The heap is managed by using new and delete (in C).

Correct!

●

Yes, stack memory is automatically allocated and cleared when a function is called. The heap is managed by using malloc and free (in C)..

○ No, they both refer to the computer's memory

Question 7

1 / 1 pts

What is a pointer?

○ it is a special variable type that exists only in C

Correct!

● it is just another variable type that stores memory addresses as their value

○ it is just another variable type

Question 8

1 / 1 pts

Which of the following results in a segfault?

Correct! ☑ dereferencing a pointer that has a NULL value

Correct! ☑ dereferencing a pointer that has not been initialized

Correct! ☑ dereferencing a pointer that refers to memory that has been freed

□ dereferencing a pointer that stores a reference to an array

Question 9

1 / 1 pts

Which of the following loops in the C programming language work like the equivalent loop in Python?

□ for loop

Correct! ☑ while loop

□ do-while loop

Question 10

1 / 1 pts

NULL has special meaning in C. Which of the following can it be used for?

Correct! ☑ Indicating the end of a linked list.

Correct! ☑ Indicating the end of a string.

□ As the return type of a function.

□ As another term for 0 in arithmetic expressions.

Quiz Score: **10** out of 10

Learning Activity: Algorithmic Foundations (classic)

Due Oct 12 at 2:59pm **Points** 10 **Questions** 10 **Available** after Oct 9 at 2am **Time Limit** None
Allowed Attempts Unlimited

Instructions

Select the *best* answer for each of the following questions.

Take the Quiz Again

Attempt History

	Attempt	Time	Score
KEPT	Attempt 2	2 minutes	10 out of 10
LATEST	Attempt 2	2 minutes	10 out of 10
	Attempt 1	1,545 minutes	4.67 out of 10

Score for this attempt: **10** out of 10

Submitted Oct 10 at 11:03pm

This attempt took 2 minutes.

Question 1

1 / 1 pts

A proof is a convincing argument that shows that something is true. It is dependent upon certain elements that we can assume to be true which are called ___.

□ invariants

□ proposition

□ predicate

Correct!

☑ definitions

Question 2

1 / 1 pts

When can you use binary search and still have an $O(\log n)$ runtime?

Correct!

● with a sorted array

○ with an unsorted array

○ with an unsorted doubly-linked list

○ with a sorted doubly-linked list

Question 3

1 / 1 pts

Which of the following is not an example of a big-Oh runtime?

Correct! ● Polynomial

○ Constant

○ Linear

○ Quadratic

Question 4

1 / 1 pts

Which of the following take 1 time step in the RAM model of computation? (select all that apply)

Correct! ☑ multipying or dividing

Correct! ☑ adding or subtracting

Correct! ☑ accessing an element in an array

□ executing a loop

Question 5

1 / 1 pts

Proof by induction requires the following 4 steps (in order):

1. Define a predicate P(n)
2. [Select]
3. [Select]
4. Invoke induction to prove P(n+1)

Answer 1:

Correct!

Define a predicate P(n)

Answer 2:

Correct!

Prove P(0) is true

Answer 3:

Correct!

Assume P(n)

Answer 4:

Correct!

Invoke induction to prove P(n+1)

Question 6

1 / 1 pts

Which of the following are steps that we should identify in an algorithm to understand if our algorithm works?

Correct! ☑ initialization

Correct! ☑ maintenance

□ predicate

Correct! ☑ termination

Question 7

1 / 1 pts

Deleting from a hashtable is ...

Correct! ● $O(1)$

○ $O(n)$

○ depends on whether it is sorted

○ $O(\log n)$

Question 8

1 / 1 pts

Which of the following best applies to an algorithm?

Correct!

● Is a set of steps to be followed to solve a problem

○ Should be run in an thread.

○ All of the others.

○ Requires a computer to run

Question 9

1 / 1 pts

Which of the following best describes *asymptotic bounding*?

Correct!

● Is a function that classifies the runtime family than an algorithm belongs to.

○ Represents the best case runtime of an algorithm

○ Represents an target runtime that we do not want an algorithm to exceed

○ Represents an approximation of runtime that is independent of input size

Question 10

1 / 1 pts

The RAM model of computation ...

○ helps us reason about algorithmic memory requirements

Correct!

● helps us reason about algorithmic performance

○ helps us reason about algorithmic correctness

○ helps us reason about the caching effects of an algorithm

Quiz Score: **10** out of 10

Learning Activity: Debugging and Assembly

Due Sep 29 at 12:30am **Points** 10 **Questions** 10 **Available** after Sep 25 at 2:30pm **Time Limit** None
Allowed Attempts Unlimited

Take the Quiz Again

Attempt History

	Attempt	Time	Score
KEPT	Attempt 3	less than 1 minute	10 out of 10
LATEST	Attempt 3	less than 1 minute	10 out of 10
	Attempt 2	2 minutes	8 out of 10
	Attempt 1	3 minutes	6.75 out of 10

Score for this attempt: **10** out of 10

Submitted Sep 28 at 1:11am

This attempt took less than 1 minute.

Question 1

1 / 1 pts

What are some features of a CISC Architecture? (Check all that apply)

☐ Generally used more often for general purpose machines like a PC, vs a specific piece of hardware (like a gaming console)

Correct! ☑ Instructions often do multiple operations

☐ Instructions do exactly one thing per operation

Correct! ☑ Generates smaller code because more operations are done at once

Question 2

1 / 1 pts

What does the MOVQ instruction do?

Correct! ● Moves an 8-byte word of data

○ Moves a 4-byte word of data

○ Moves a 2-byte word of data

○ Moves a 1-byte word of data

Question 3

1 / 1 pts

What are the best practices for disassembling a binary and reading the source code? (Check all that apply)

Correct! ☑ Disable optimizations and other code transformations

Correct! ☑ Using the disassembly (-d) flag when using objdump

Correct! ☑ Make sure you compiled with debugging symbols

Question 4

1 / 1 pts

Which of the following registers represent a 4-byte word?

Correct! ● %EAX

○ %AX

○ %RAX

○ %AL

Question 5

1 / 1 pts

What does the MOVB instruction do?

○ Moves an 8-byte word of data

○ Moves a 2-byte word of data

○ Moves an 4-byte word of data

Correct!

● Moves a 1-byte of data

Question 6

1 / 1 pts

Which of the following commands can be used in GDB to execute a function call without stepping into it?

○ run

Correct!

● n

○ s

○ b

Question 7

1 / 1 pts

What is x86-64 assembly? (Check all that apply)

□ a 32-bit instruction set for 64-bit architectures

Correct! ☑ a 64-bit instruction set for x86 architectures

Correct! ☑ a CISC artchitecture

Correct! ☑ a human-readable form of machine code made popular by Intel

Question 8

1 / 1 pts

Why does the compilation toolchain generate assembly?

□ Because assembly was needed at one time and we just haven't taken it out of the toolchain

Correct! ☑ To make C portable to different operating systems and platforms

Correct! ☑ To provide a human-readable form of machine code

□ No answer text provided.

Question 9

1 / 1 pts

Which of the following flags should you use if you are going to use gdb to debug your program?

○ -d

○ -o

Correct!

● -g

○ -c

Question 10

1 / 1 pts

Why might a programmer want to learn assembly? (Check all that apply)

Correct!

☑ To learn how to optimize their code and if it really needs to be optimized

Correct!

☑ To help find and locate bugs in software

Correct! ☑ To reverse engineer software for good!

Correct! ☑ To understand and implement compilers

Quiz Score: **10** out of 10

Learning Activity: Divide & Conquer (classic)

Due Oct 19 at 2:59pm **Points** 10 **Questions** 10 **Available** after Oct 16 at 2am **Time Limit** None
Allowed Attempts Unlimited

Take the Quiz Again

Attempt History

	Attempt	Time	Score
KEPT	Attempt 5	2 minutes	10 out of 10
LATEST	Attempt 5	2 minutes	10 out of 10
	Attempt 4	1 minute	9 out of 10
	Attempt 3	2 minutes	9 out of 10
	Attempt 2	2 minutes	9 out of 10
	Attempt 1	3,963 minutes	9 out of 10

Score for this attempt: **10** out of 10
Submitted Oct 19 at 12:53am
This attempt took 2 minutes.

Question 1

1 / 1 pts

Which best describes a *randomized algorithm*?

Correct!

● usually have complexity expressed in terms of their average case, not their worst case

○ usually have a constant complexity which is why randomized algorithms are used

○ usually do not have a time complexity due to their random behavior

○ are hard to debug because of race conditions that result from their nondeterministic behavior

Question 2

1 / 1 pts

There are many sorting algorithms because ...

○ the best sorting algorithm depends on the number of swaps it must perform

○ the best sorting algorithm depends on how easy it is to implement

○ the best sorting algorithm depends on the size of the input

Correct!

● the best sorting algorithm depends on the nature of the problem

Question 3

1 / 1 pts

Which of the following are true with respect to n-squared sorting algorithms? (Select all that apply)

☐ Get their runtime because of the number of times we need to swap values

☐ Always use a nested loop structure

☐ Can be optimized to give a better runtime by visiting each element only once.

Correct! ☑ Get their runtime because for every element, we may have to compare it to all of the others

Question 4

1 / 1 pts

Which of the following best describes a *divide and conquer* algorithm?

Correct! ● can be analyzed by examining the resulting recurrence tree

○ all of the others

○ rely on recursion in their implementation

○ have an $O(n \log n)$ runtime

Question 5

1 / 1 pts

Which of the following is an advantage that insertion sort has over merge sort?

○ None of the others

Correct!

● It can be done *in place*

○ It only has to swap element once

○ It has a better runtime

Question 6

1 / 1 pts

When you see a $2T(\frac{n}{2})$ in a recurrence, you can be sure that ...

Correct!

● all of the others

○ the term will contribute $\log n$ to the overall complexity of the algorithm

○ there is most likely recursion involved

○ the input set gets divided into 2 parts

Question 7

1 / 1 pts

Which best describes the *substitution method*?

○ only works on recurrences that follow a particular pattern

Correct!

● systematic approach for proving complexity expressed as a recurrence

○ technique for substituting one sort for a better sort

○ technique for avoiding a stack overflow

Question 8

1 / 1 pts

Which of the following relates to implementing recursion in C?

○ is an alternative way to implement a loop

○ can be optimized into a loop when written in a tail-recursive way

Correct! ● all of the others.

○ adds to the execution (or call) stack until the base case is reached

Question 9

1 / 1 pts

Which of the following is true about merge sort?

Correct! ☑ always results in a tree that has $\log_2 n$ levels

Correct! ☑ does $O(n)$ work at each level in the tree

□ merges the elements around a pivot

□ can be implemented *in place*

Question 10

1 / 1 pts

A *recurrence tree proof* is ...

○ not a proof, but a way to think about the algorithm complexity of merge sort

○ a proof, but it only works for merge sort

○ a proof, but only works for binary search

Correct!

● a proof that can be formalized to the whole family of *divide and conquer* algorithms

Quiz Score: **10** out of 10

Learning Activity: Dynamic Programming

Due Dec 3 at 3:59pm **Points** 10 **Questions** 10 **Available** after Nov 20 at 3:30am **Time Limit** None
Allowed Attempts Unlimited

Instructions

For each question, select the *best* answer.

Take the Quiz Again

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	3 minutes	10 out of 10

Score for this attempt: **10** out of 10

Submitted Nov 27 at 10:38pm

This attempt took 3 minutes.

Question 1

1 / 1 pts

When is it a good idea to consider using *dynamic programming*?

○ whenever algorithms have a greedy solution

Correct! ● whenever algorithms are expensive and use a lot of recomputation

○ whenever algorithms can benefit from caching

○ whenever algorithms are recursive

Question 2

1 / 1 pts

Which of the following best describes *memoization*?

○ eliminate iteration and just use recursion

○ eliminate look up values and just recompute values

Correct! ● eliminate recomputation and just look up values

○ eliminate recursion and just use iteration

Question 3

1 / 1 pts

Which of the following are we referring to when we talk about the space-time tradeoff?

○ depends on the algorithm

Correct! ● time savings

○ space savings

○ both space and time savings

Question 4

1 / 1 pts

Which of the following best describes the *longest common substring* problem?

Correct! ● Uses $O(n * m)$ time, for strings of different lengths

○ Uses an adjacency matrix to convert strings into a graph

○ Uses $O(n)$ time for strings of the same length

○ Uses $O(n^2)$ time, for strings of different lengths.

Question 5

1 / 1 pts

Which of the following are used by a *dynamic programming* approach? (Select all that apply)

□ memorization of common paths

□ pointers to a lot of dynamically allocated memory

Correct! ☑ memoization of computed values

□ priority queue storing the next best choice

Question 6

1 / 1 pts

Which of the following best describes exponential growth?

○ computation is will have $O(n!)$ complexity

○ computation will have $O(n^2)$ complexity

Correct! ● computation will have $O(2^n)$ complexity

○ it doesn't matter, because it will take too long to run

Question 7

1 / 1 pts

Which of the following best describes the naive recursive Fibonacci implementation?

Correct!

● requires a lot of recomputation, resulting in exponential runtime

○ requires a lot of recursive calls resulting in a large execution stack and exponential space

○ requires a lot of memory dynamically allocated memory that must be freed

○ requires a lot of recomputation, resulting in linear runtime

Question 8

1 / 1 pts

Which of the following is required to implement caching?

○ memory in the processor

○ memory on the execution stack

Correct!

● memory of any kind (static or dynamic)

○ disk storage

Question 9

1 / 1 pts

Which of the following algorithms benefit from *dynamic programming?*

○ all divide and conquer algorithms

○ binary search

○ merge sort

Correct!

● Fibonacci numbers

Question 10

1 / 1 pts

Which of the following best describes *dynamic programming* strategy?

Correct!

● divides problem and solves the smaller version

○ solves the problem using traversal strategies

○ solves the problem using the best possible local choice

○ solves the problem using brute force

Quiz Score: **10** out of 10

Learning Activity: Graphs

Due Nov 2 at 2:59pm **Points** 10 **Questions** 10 **Available** after Oct 30 at 2:30am **Time Limit** None
Allowed Attempts Unlimited

Instructions

Answer each of the following questions.

Take the Quiz Again

Attempt History

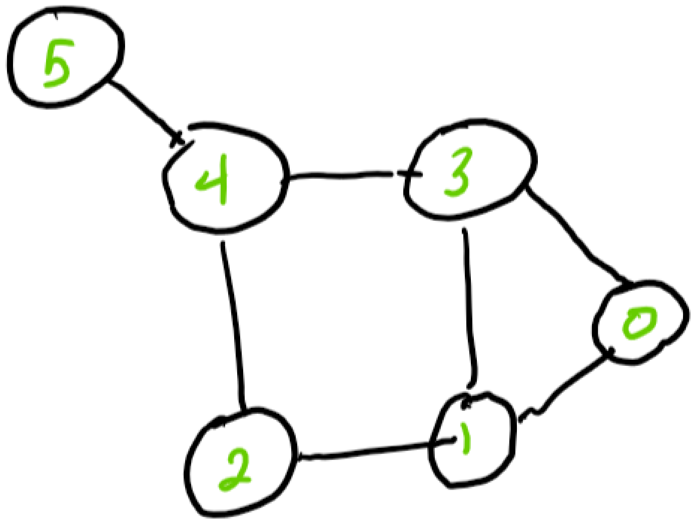
	Attempt	Time	Score
KEPT	Attempt 4	2 minutes	10 out of 10
LATEST	Attempt 4	2 minutes	10 out of 10
	Attempt 3	4 minutes	9 out of 10
	Attempt 2	5 minutes	8.25 out of 10
	Attempt 1	6 minutes	5 out of 10

Score for this attempt: **10** out of 10
Submitted Oct 30 at 9:15pm
This attempt took 2 minutes.

Question 1

1 / 1 pts

Suppose we do a *depth-first* traversal of the following graph starting at 0, Assume that neighbors of a node are processed in increasing numerical order. What is the order that the nodes are visited?



Correct!

● 0, 1, 2, 4, 3, 5

○ 0, 1, 3, 2, 4, 5

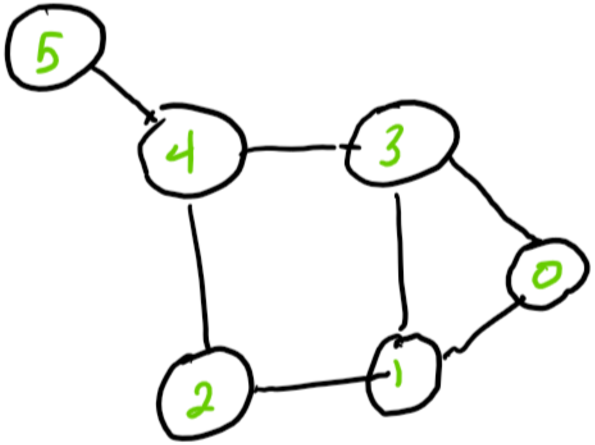
○ 0, 1, 2, 3, 4, 5

○ 0, 3, 4, 5, 2, 1

Question 2

1 / 1 pts

Select which represent a valid cycle in the following graph.



□ 1 -> 2 -> 3 -> 4 -> 1

Correct!

☑ 0 -> 1 -> 3 -> 0

□ 0 -> 1 -> 2 -> 4 -> 0

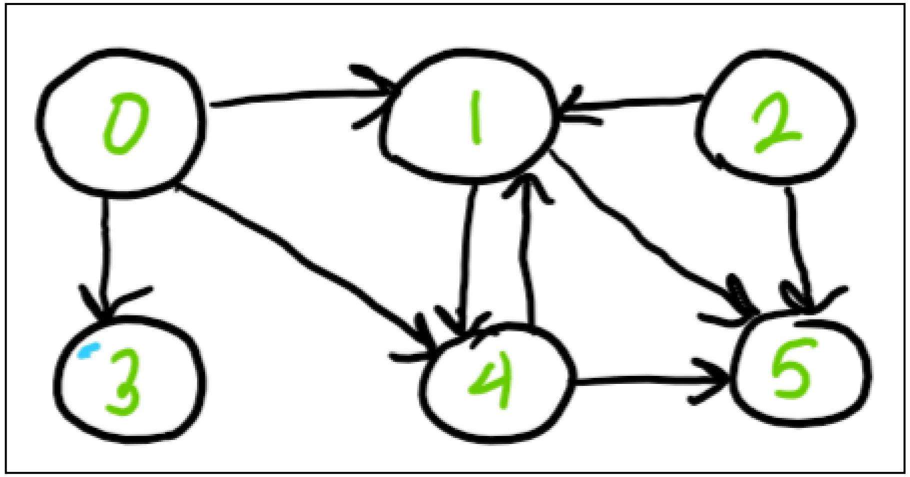
Correct!

☑ 5 -> 4 -> 3 -> 1 -> 2 -> 4 -> 5

Question 3

1 / 1 pts

Select which represent a valid path in the following graph.



Correct! ☑ 4 -> 1 -> 4 -> 5

□ 0 -> 1 -> 2 -> 5

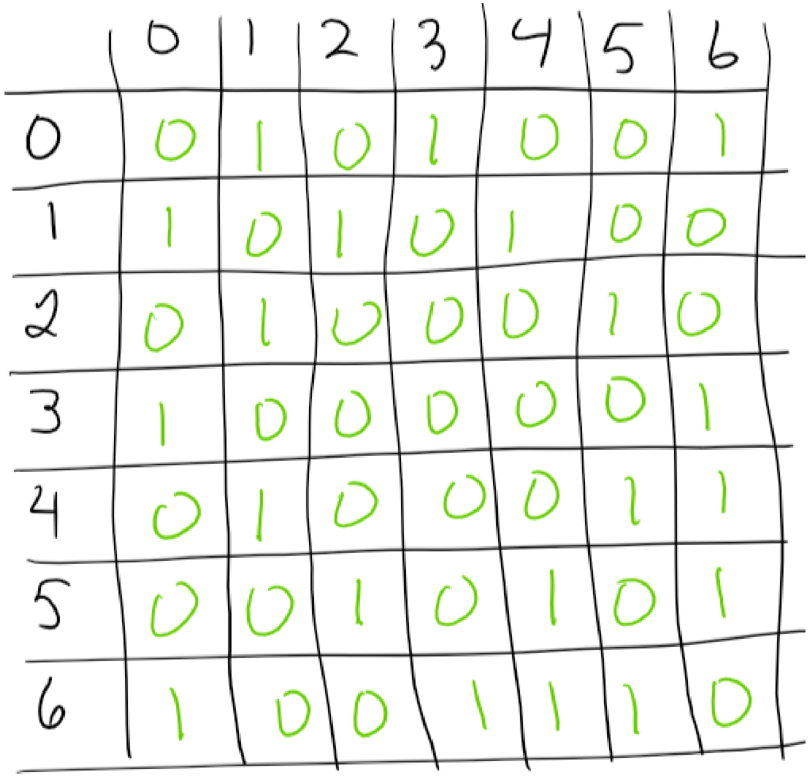
Correct! ☑ 0 -> 1 -> 4 -> 5

Correct! ☑ 0 -> 4 -> 1 -> 5

Question 4

1 / 1 pts

Given the following an undirected graph represented as an adjacency matrix, which of the following is a valid path. Select all that apply.



	0	1	2	3	4	5	6
0	0	1	0	1	0	0	1
1	1	0	1	0	1	0	0
2	0	1	0	0	0	1	0
3	1	0	0	0	0	0	1
4	0	1	0	0	0	1	1
5	0	0	1	0	1	0	1
6	1	0	0	1	1	1	0

□ 0 -> 1 -> 2 -> 3

Correct! ☑ 3 -> 0 -> 1 -> 2

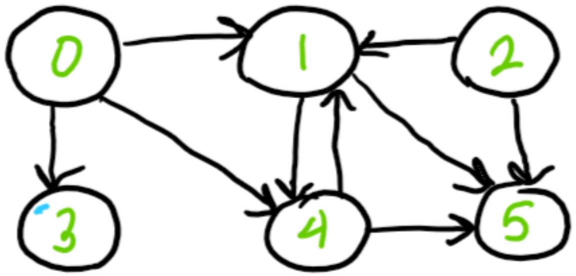
Correct! ☑ 6 -> 5 -> 4 -> 6

□ 1 -> 2 -> 4 -> 3

Question 5

1 / 1 pts

Suppose we do a *breadth-first* traversal of the following graph starting at node 0. Assume that neighbors of a node are processed in increasing numerical order. What is the order that the nodes are visited?



○ 0, 1, 4, 5, 3, 2

○ 0, 1, 3, 4, 5, 2

○ 0, 1, 2, 3, 4, 5

Correct!

● 0, 1, 3, 4, 5

Question 6

1 / 1 pts

Which of the following do you think can be represented in a graph? Select all that apply.

Correct!

☑ file system where vertices are the files and directory while edges show the hierarchical relationships between them

Correct!

☑ people relationships where vertices are people while edges show relationships between them

Correct!

☑ airline routes where vertices are airports while edges show the flights between them

Correct!

☑ memory of a program where vertices are variables (including structs) while edges show pointers between them

Question 7

1 / 1 pts

Why do we need to keep track of which nodes are visited (or discovered) when we are implementing a graph traversal?

○ to avoid implementing the traversal recursively

Correct!

● to ensure that we do traverse each node exactly one time

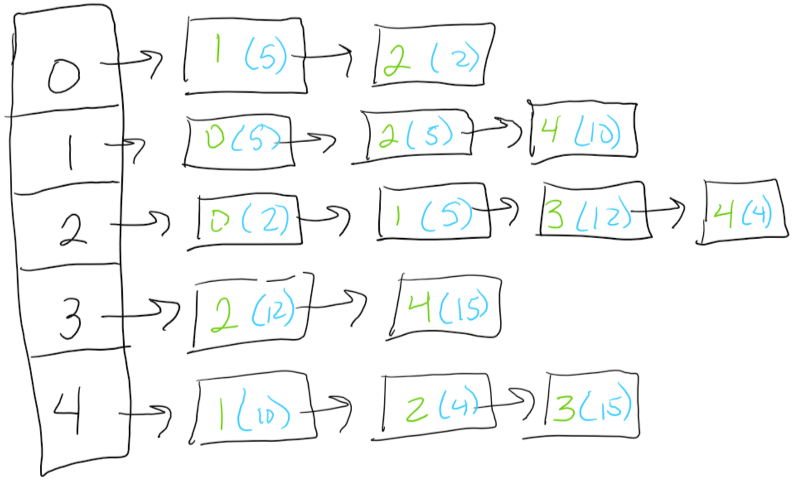
○ to detect when a graph has cycles

○ to make the algorithm different from traversing a tree

Question 8

1 / 1 pts

Given the following weighted (displayed in parentheses), directed graph represented as and adjacency list, which of the following is a valid path. Select all that apply.



Correct!

☑ 0 -> 1 -> 2 -> 3

□ 3 -> 4 -> 0 -> 1

Correct!

☑ 1 -> 2 -> 3 -> 4

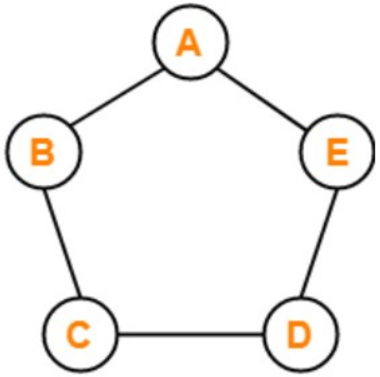
Correct!

☑ 0 -> 1 -> 2 -> 3

Question 9

1 / 1 pts

Let C_n be an undirected graph on *n* vertices whose edges form a cycle. For example, C_5 would be



Suppose you want to perform several algorithms on the graph $C_{1000000000}$. Which representation would be best to use to implement this graph?

○ Adjacency matrix representation

○ Either representation is fine for this as they would provide similar time and space complexities.

Correct!

● Adjacency list representation

Question 10

1 / 1 pts

Breadth-first traversals of a graph

○ can be implemented recursively to avoid using a stack

○ use a stack to store the vertices that have not been traversed

Correct! ● use a queue to store the vertices that have not been traversed

○ use a queue to store the vertices that have been traversed

Quiz Score: **10** out of 10

Learning Activity: Greedy

Due Nov 9 at 3:59pm **Points** 10 **Questions** 10 **Available** after Nov 6 at 2:30am **Time Limit** None
Allowed Attempts Unlimited

Instructions

For each of the following questions, select the ***best*** answer.

Take the Quiz Again

Attempt History

	Attempt	Time	Score
KEPT	Attempt 2	2 minutes	10 out of 10
LATEST	Attempt 2	2 minutes	10 out of 10
	Attempt 1	10 minutes	10 out of 10

Score for this attempt: **10** out of 10
Submitted Dec 6 at 1:15pm
This attempt took 2 minutes.

Question 1

1 / 1 pts

Which of the following represents the *base case* of the proof by induction for the interval scheduling problem?

○ the empty set

Correct! ● a meeting set with 1 element

○ time of the first element in the optimal set

○ time of the first element of the set defined by the selection choice

Question 2

1 / 1 pts

Which of the following greedy strategies works best for the *interval scheduling problem*?

○ fewest interruptions

○ shortest interval

○ starts first

Correct! ● ends first

Question 3

1 / 1 pts

Which of the following is an assumption that is made by the *proof by contradiction* for the interval scheduling problem?

○ Start times were better in the optimal algorithm

Correct! ● Greedy algorithm is a contradiction

○ Greedy algorithm is optimal

○ Optimal algorithm is not greedy

Question 4

1 / 1 pts

Which of the following affects the runtime of Dijkstra's?

Correct! ☑ the representation of the graph itself

□ the number of iterations of the loop

Correct! ☑ the cost of finding the node with the minimum distance from the set of visited nodes

□ the number of recursive calls made

Question 5

1 / 1 pts

Which of the following graphs can you apply Dijkstra's algorithm to?

Correct! ☑ undirected graph with non-negative edge weights

Correct! ☑ directed graph with non-negative edge weights

□ directed graph with negative edge weights

□ undirected graph with negative edge weights

Question 6

1 / 1 pts

Which of the following best describes when Dijkstra's Algorithm stops?

○ when we have completed a DFS of the graph

Correct! ● when we have completed a BFS of the graph

○ when we find the shortest path

○ when we have looked at all the edges

Question 7

1 / 1 pts

Which of the following best describes a *greedy algorithm*?

○ All of the others

○ Similar to divide and conquer, they take steps towards making the problem smaller

Correct!

● Work by always choose the "best" option at each step

○ Always generate a unique, correct solution

Question 8

1 / 1 pts

Which of the following is the first step of Dijkstra's algorithm?

○ Initialize distance of all the nodes to -1, except the source which has a distance of infinity

○ Initialize distance of all the nodes to infinity

Correct! ● Initialize distances of all nodes to infinity, except the source which has a distance of 0

○ Initialize distances of all nodes to infinity, except the source which has a distance of -1

Question 9

1 / 1 pts

Dijkstra's Algorithms computes ...

○ path with smallest average cost per edge

○ a path that does not involve cycles

○ shortest path from starting node to the destination node

Correct! ● shortest path from a starting node to every node in the grapah

Question 10

1 / 1 pts

Which of the following best describes the *shortest path* between two nodes in a directed, weighted graph?

○ path with the smallest average cost per edge

○ path with the smallest number of edges

Correct!

● path with the smallest sum of the weights of the edges in the path

○ path that does not contain cycles

Quiz Score: **10** out of 10

Learning Activity: Trees

Due Oct 26 at 2:59pm **Points** 10 **Questions** 10 **Available** after Oct 23 at 2am **Time Limit** None
Allowed Attempts Unlimited

Instructions

Select the *best* answer for each of the following questions.

Take the Quiz Again

Attempt History

	Attempt	Time	Score
KEPT	Attempt 7	less than 1 minute	10 out of 10
LATEST	Attempt 7	less than 1 minute	10 out of 10
	Attempt 6	less than 1 minute	9.67 out of 10
	Attempt 5	less than 1 minute	8.33 out of 10
	Attempt 4	less than 1 minute	9 out of 10
	Attempt 3	3 minutes	8 out of 10
	Attempt 2	2 minutes	6 out of 10
	Attempt 1	5 minutes	4 out of 10

Score for this attempt: **10** out of 10

Submitted Oct 23 at 10:42am

This attempt took less than 1 minute.

Question 1

1 / 1 pts

The reason why we print recursively is:

○ to provide a way to handle printing a tree in different orders

○ to avoid using a loop

Correct!

● to avoid using more memory required by an auxiliary data structure

○ because we can implement using tail recursion

Question 2

1 / 1 pts

Why did we need the `addHelper` function when implementing `add`?

○ to make add more efficient

○ to handle the creation of the node

○ to make room for inserting a node in the middle of the tree

Correct! ● to implement add recursively

Question 3

1 / 1 pts

The add function for trees ...

○ is O(1)

○ All of the others

○ always adds a leaf node

Correct! ● first checks the root to see if it is null

Question 4

1 / 1 pts

A *breadth-first search* traversal ...

○ is implemented recursively

Correct! ● visits each node while keeping track of levels

○ has a time complexity that is bounded by the height of the tree

○ starts at the leaves and moves up through the tree

Question 5

1 / 1 pts

Which of the following are characteristics of a *heap*?

Correct! ☑ the value in the parent is greater than the value in all of its children

Correct! ☑ a binary tree

Correct! ☑ is complete

□ value of left child < value of parent < value of right child

Question 6

1 / 1 pts

Which of the following is a property of a *complete* tree?

○ leaf nodes are all are all on the last level

○ last level has all left children

○ all levels, except the last one, are not full

Correct! ● last level has all nodes to the left

Question 7

1 / 1 pts

Children in a tree ...

○ are created when a new node is forked.

○ can be accessed in O(1) time

○ are all leaves

Correct! ● are accessed recursively

Question 8

1 / 1 pts

Operations on a *binary search tree* (BST) include:

□ adding to the front

Correct! ☑ searching for an element

□ sorting the elements in O(n log n) time

□ removing an element using an index

Question 9

1 / 1 pts

Which of the following is true about nodes in a *binary tree*?

□ can have 0 or 2 children

Correct! ☑ can have up to 2 children

□ can have up to 2 siblings

□ can have up to 2 parents

Question 10

1 / 1 pts

Which of the following *best* describes the Tree data structure?

○ Trees represent a more complicated way of storing data

Correct!

● Trees represent a data hierarchy

○ Trees are very similar to a doubly linked list because it has two pointers to other nodes

○ Trees are good for storing data sequentially

Quiz Score: **10** out of 10