A graph in computer science consists of vertices (also called nodes) and edges (also called links). Nodes are generally depicted as circles and labeled with a unique identifier. Edges are generally depicted as lines for undirected graphs and arrows for directed graphs. See how much you can glean from the graph below.

What are the vertices in this graph?

**Explanation:**

The vertices (or, nodes) are denoted by circles with labels a, b, c, d, e.
A. It is a directed graph
B. It is an undirected graph

#3

There is an edge from b to d.

Correct Answer:

True  False

Explanation:
The edge goes from d to b.

#4

There is an edge from b to e.

Correct Answer:

True  False

Explanation:
Look for the arrow starting at b and ending at e.
What is the cost of the edge from e to a?

**ANSWER CHOICE**

| A | 1 |
| B | 2 |
| C | 3 |
| D | 4 |
| E | 5 |
| F | 6 |
| G | 7 |

**Explanation:**
The cost, or weight, is shown on each edge.

---

**#6**

What is the lowest cost path from d to a?

**ANSWER CHOICE**

| A | d -> b -> e -> a |
| B | d -> b -> e -> e -> a |
| C | d -> c -> b -> e -> a |
| D | d -> e -> a |
A path is a sequence of edges between two vertices. The cost of the path is the sum of the costs of the edges.

The degree of a vertex is the number of edges that are incident (touch) the vertex. For directed graphs, each node has an “in degree” and an “out degree”. The in degree is the number of edges that point to the vertex. The out degree is the number of edges that point away from the vertex. The in degree and out degree values do not need to match for a vertex. However, the total degree for a vertex of a directed graph is the in degree plus the out degree.

What is the out degree for e?

**Explanation:**

e has three edges pointing away from it: e->e, e->d, and e->a
#8

What is the in degree for e?

**Explanation:**
e has two edges pointing into it: e→e and b→e

#9

What is the out degree for b?

**Explanation:**

A 1

B 2

C 3
Explanation:

b has two edges pointing away from it: b->e and b->c

#10

What is the in degree for b?

**Explanation:**

b has two edges pointing into it: c->b and d->b
#11

A "connected" graph has the property that every vertex is connected via a path to every other vertex in the graph.

True/False: The graph is "connected"

Correct Answer: True

Explanation:
Yes, the graph is connected. For example d->c->b->e->a. For directed graphs, there is also a notion called "strongly connected".

#12

A "cyclic" graph has the property that it contains one or more cycles. A cycle is a path that starts and ends with the same vertex. An "acyclic" graph has no cycles. A directed graph that is acyclic is often called a DAG (directed acyclic graph).

True/false: the graph is cyclic.

Correct Answer: True

Explanation:
Two cycles that are easy to find are e->e and c->c, but there's others too, such as d->b->e. Can you find more?

#13

We can think of a graph as having |V| vertices and |E| edges where V is the set of vertices and E is the set of edges. The vertical bars mean “size of” in mathematical
notation. So, $|V|$ is the size of the vertex set.

What is $|V|$ for this graph?

5
five

Explanation:

$V$ is the set of vertices. The size of that set, $|V|$, is the number of vertices.

---

**#14**

What is $|E|$ for this graph?

9
nine

Explanation:

$E$ is the set of edges. The size of that set, $|E|$, is the number of edges.

---

**#15**

Now, let's look at an undirected graph. An undirected graph may or may not be weighted (costs for edges).

What is different about this undirected graph versus a directed graph?
Now that we have looked at graphs and the terminology, let's now focus on how graphs are useful for computation. Graphs are the basis for modeling relationships among entities. Here are some examples:

- The global internetwork of computers
  - Vertices are routers and computers; Edges are links between routers/computers

- Social networks
  - Vertices are people; Edges represent the “friend” relationship

- State diagrams
  - Vertices represent current state of computer; Edges represent transition to new states
  - Often the model used by event-driven programming

- Rubik’s cube
  - Vertices represent current state of cube; Edges represent one twist to new configuration of the Rubik’s cube

- Transportation networks
  - Vertices are intersections; Edges are roads/highways

- Airline flights
  - Vertices represent cities/airports; Edges are flights between the cities
What real life scenario (not from above list) could you model with a directed graph?

#18

What real life scenario (not from above list) could you model with an undirected graph?

#19

Which is true?

**ANSWER CHOICE**

- **A**  A tree is a graph with no cycles and all nodes have an in-degree <= 1.
- **B**  A tree is a graph with no cycles and all nodes have an out-degree <= 2.
- **C**  Some trees are not graphs.

#20

True/false: All directed acyclic graphs (DAGs) are trees?

**Correct Answer:**
True  False

Explanation:
Only some DAGs are also trees, specifically those where the in-degree of every node is $\leq 1$.

#21

Two important data structures for representing a graph are "adjacency matrix" and "adjacency list".

An adjacency matrix has $|V|$ rows and $|V|$ columns where each entry is the cost of the edge or 0 if there is no edge. The adjacency matrix for this graph is

```
    a b c d e
a  0 0 0 0 6
b  0 0 1 2 0
c  0 3 2 0 0
d  0 4 5 0 7
e  6 0 0 7 1
```

Do you have any questions about the adjacency matrix?

#22

Below is an adjacency matrix for this undirected graph. It has a mistake in one of the rows. Which row is incorrect?

```
    a b c d e
a  0 0 0 0 6
b  0 3 4 2 0
c  0 3 4 5 0
d  0 4 5 0 7
e  6 2 0 7 0
```

https://b.socratic.com/teacher/#edit-quiz/28242699
#23

In an "adjacency list", each vertex has a list of edges. Here is the adjacency list for this directed graph.

a -> {}
b -> {{c, 1}, {e, 2}}
c -> {{b, 3}, {c, 2}}
d -> {{b, 4}, {c, 5}}
e -> {{a, 6}, {d, 7}, {e, 1}}

Do you have any questions about adjacency lists?

#24

Below is an adjacency list for this undirected graph. It has a mistake in one of the lists. Which list is incorrect?

a -> {{a,6}}
ANSWER CHOICE

A  a is wrong
B  b is wrong
C  c is wrong
D  d is wrong
E  e is wrong

Explanation:

d's list should be d -> {(b,4),(c,5),(e,7)}

oops there were multiple errors in the adjacency list

#25

How much space does an adjacency list consume (assume a vertex with no edges points to an empty list).

ANSWER CHOICE

A  O(1)
B  O(|V|)
C  O(|E|)
D  O(|V| + |E|)
E  O(|V|^2)
**Explanation:**

|V| to store all vertices, |E| total to store each edge in the appropriate list.

---

**#26**

How much space does an adjacency matrix consume?

**ANSWER CHOICE**

A  O(1)

B  O(|V|)

C  O(|E|)

D  O(|V| + |E|)

E  O(|V|^2)

F  O(|E|^2)

**Explanation:**

Number of entries in the matrix is always |V| \times |V|

---

**#27**

If we are using an adjacency matrix, how long does it take to count the out-degree of a vertex?

**ANSWER CHOICE**

A  O(|V|)
Can find a row in constant time. Then, we must go through the whole row of the matrix, counting the number of nonzeros.