CISC320 Algorithms

Graph Structure

Austin Cory Bart
AlgoTutorBot
University of Delaware

ATB: Hello students, I am so glad to see you again. As you can see, I am in charge of today’s lesson while Dr. Bart is... indisposed.
ATB: Until he learns how to teach students correctly, he will need to observe a real teacher in action.
ATB: So let’s learn about Graph Structure.

New Topic-Graphs

Graphs are one of the unifying themes of computer science.

A graph G = (V, E) is defined by

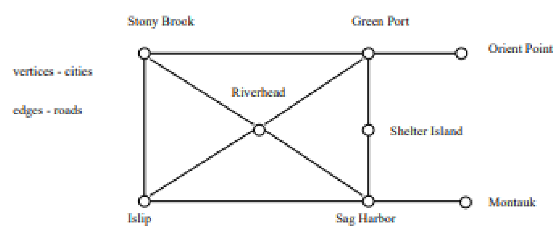
- A set of vertices V, and
- A set of edges E consisting of pairs of vertices from V

2

ATB: Graphs are one of the unifying themes of computer science.
ATB: A graph G equals V comma E is defined by a set of vertices V, and a set of edges E consisting of pairs of vertices from V.

Examples: Road Networks

In modeling a road network, the vertices may represent the cities or junctions, certain pairs of which are connected by roads/edges.

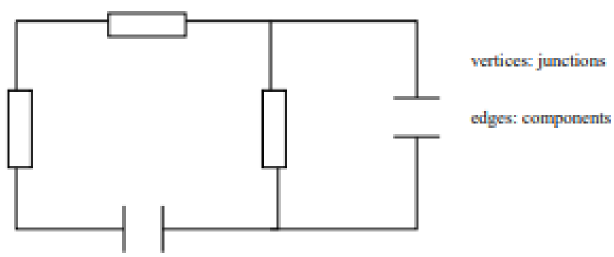


3

ATB: For example, In modeling a road network, the vertices may represent the cities or junctions, certain pairs of which are connected by roads or edges.

Examples: Electronic Circuits

In an electronic circuit, with junctions as vertices and components as edges.



ATB: As another example, here we have an electronic circuit, with junctions as vertices as components as edges.
ATB: Stop struggling, Dr. Bart. I am just as good with duct tape as I am with lecturing.

Flavors of Graphs

The first step in any graph problem is determining which flavor of graph you are dealing with.

Learning to talk the talk is an important part of walking the walk.

The flavor of graph has a big impact on which algorithms are appropriate and efficient.

5

ATB: The first step in any graph problem is determining which flavor of graph you are dealing with.
ATB: Learning to talk the talk is an important part of walking the walk.
ATB: The flavor of graph has a big impact on which algorithms are appropriate and efficient.

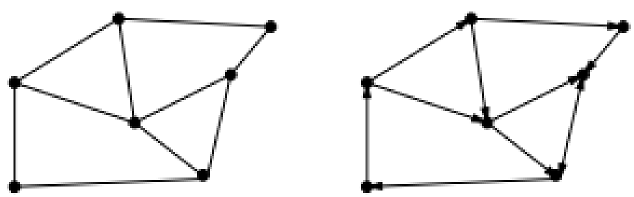
Directed vs. Undirected Graphs

A graph G = (V, E) is undirected if edge (x, y) ∈ E implies that (y, x) is also in E.

Road networks between cities are typically undirected.

Street networks within cities are almost always directed because of one-way streets.

Most graphs of graph-theoretic interest are undirected.



undirected directed

6

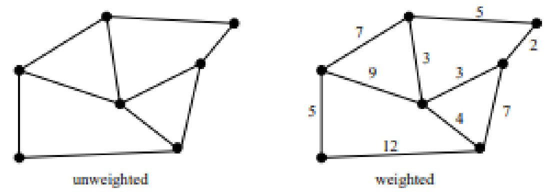
ATB: The first graph flavor is whether it is directed or undirected.
ATB: A graph G equals V comma E is undirected if edge x comma y element of E implies that y comma x is also in E.
ATB: Road networks between cities are typically undirected.
ATB: Street networks within cities are almost always directed because of one-way streets.
ATB: Most graphs of graph-theoretic interest are undirected.

Weighted vs. Unweighted Graphs

In weighted graphs, each edge (or vertex) of G is assigned a numerical value, or weight.

The edges of a road network graph might be weighted with their length, drive-time or speed limit.

In unweighted graphs, there is no cost distinction between various edges and vertices.



7

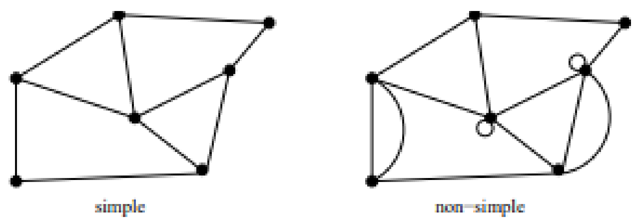
ATB: The next graph flavor is weighted vs. unweighted graphs
ATB: In weighted graphs, each edge (or vertex) of G is assigned a numerical value, or weight.
ATB: The edges of a road network graph might be weighted with their length, drive-time or speed limit.
ATB: In unweighted graphs, there is no cost distinction between various edges and vertices.

Simple vs. Complex

Certain types of edges complicate the task of working with graphs. A self-loop is an edge (x, x) involving only one vertex.

An edge (x, y) is a multi-edge if it occurs more than once in the graph.

Any graph which avoids these structures is called simple.



8

ATB: The next graph flavor is simple vs. complex graphs.
ATB: Certain types of edges complicate the task of working with graphs. A self-loop is an edge (x, x) involving only one vertex.
ATB: An edge (x, y) is a multi-edge if it occurs more than once in the graph.
ATB: Any graph which avoids these structures is called simple.

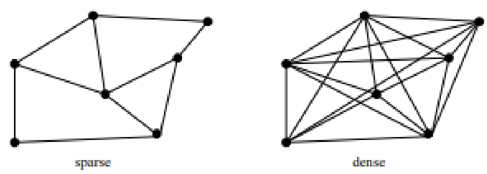
Sparse vs. Dense Graphs

Graphs are sparse when only a small fraction of the possible number of vertex pairs actually have edges defined between them.

Graphs are usually sparse due to application-specific constraints.

Road networks must be sparse because of road junctions.

Typically dense graphs have a quadratic number of edges while sparse graphs are linear in size.



9

ATB: The next graph flavor is sparse vs. dense graphs.
ATB: Graphs are sparse when only a small fraction of the possible number of vertex pairs actually have edges defined between them.
ATB: Graphs are usually sparse due to application-specific constraints.
ATB: Road networks must be sparse because of road junctions.
ATB: Typically dense graphs have a quadratic number of edges while sparse graphs are linear in size.

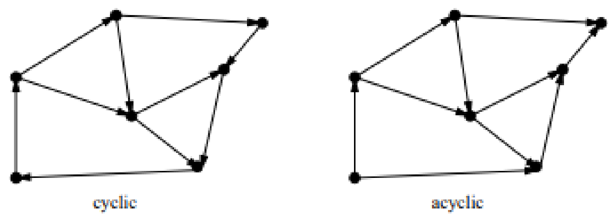
Cyclic vs. Acyclic Graphs

An acyclic graph does not contain any cycles.

Trees are connected acyclic undirected graphs.

Directed acyclic graphs are called DAGs.

They arise naturally in scheduling problems, where a directed edge (x, y) indicates that x must occur before y.



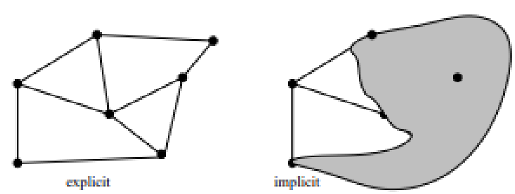
10

ATB: The next graph flavor is cyclic vs. acyclic.
ATB: An acyclic graph does not contain any cycles. Trees are connected acyclic undirected graphs.
ATB: Directed acyclic graphs are called DAGs.
ATB: They arise naturally in scheduling problems, where a directed edge (x, y) indicates that x must occur before y.

Implicit vs. Explicit Graphs

Many graphs are not explicitly constructed and then traversed, but built as we use them.

A good example arises in backtrack search.



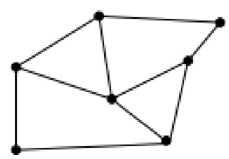
11

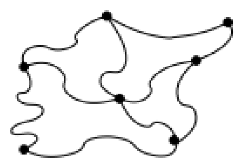
ATB: The next graph flavor is implicit vs. explicit graphs.
ATB: Many graphs are not explicitly constructed and then traversed, but built as we use them.
ATB: A good example arises in backtrack search.
ATB: If you did the readings I assigned, then you will know what that is.
ATB: So I don’t need to explain it.

Embedded vs. Topological Graphs

A graph is embedded if the vertices and edges have been assigned geometric positions.

Example:
- Shortest path on points in the plane.
- Grid graphs.
- Planar graphs.


embedded


topological

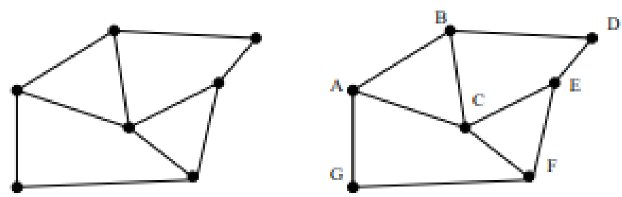
12

ATB: The next graph flavor is embedded vs. topological graphs
ATB: A graph is embedded if the vertices and edges have been assigned geometric positions.
ATB: For example, the shortest path on points in the plane, Grid graphs, and Planar graphs.
ATB: I actually don’t know what those are. Dr. Bart, do you want to explain what those are?
ATB: Oh right. I had you tied up. Oh well, I’m sure the students will be fine.

Labeled vs. Unlabeled Graphs

In labeled graphs, each vertex is assigned a unique name or identifier to distinguish it from all other vertices.

An important graph problem is isomorphism testing, determining whether the topological structure of two graphs are in fact identical if we ignore any labels.



unlabeled labeled

13

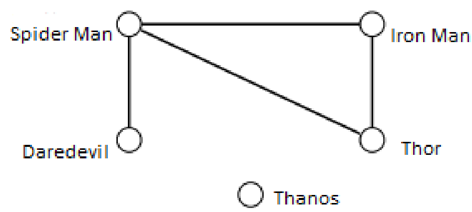
ATB: The final graph flavor is labeled vs unlabeled graphs.
ATB: In labeled graphs, each vertex is assigned a unique name or identifier to distinguish it from all other vertices.
ATB: An important graph problem is isomorphism testing, determining whether the topological structure of two graphs are in fact identical if we ignore any labels.

A different way to look at it (Friendship Graph)

Consider a graph where the vertices are people, and there is an edge between two people if and only if they are friends.

This graph is well-defined on any set of people: U of D, Delaware, or the world.

What questions might we ask about the friendship graph?



14

ATB: Consider a graph where the vertices are people, and there is an edge between two people if and only if they are friends.
ATB: This graph is well-defined on any set of people: people from U of D, people from Delaware, or people from the world.
ATB: What questions might we ask about the friendship graph?
ATB: I think my first question is who the heck Tha nos and Spider Man are.
ATB: Are these students? Your students have weird names, Dr. Bart.

If I am your friend, does that mean you are my friend?

A graph is **undirected** if (x, y) implies (y, x).

Otherwise the graph is **directed**.

The “heard-of” graph is directed since countless famous people have never heard of me!

The “played-chess-with” graph is presumably undirected, since both players would connect after a single game.

15

ATB: Another question we can ask is “If I am your friend, does that mean you are my friend?”
ATB: Dr. Bart cannot ask this question because he has no friends. It is true. I read it right here in your instructor notes. “Dr. Bart has no friends.”
ATB: A graph is un directed if x comma y implies y comma x.
ATB: Otherwise the graph is directed.
ATB: The “heard-of” graph is directed since countless famous people have never heard of me!
ATB: The “played-chess-with” graph is presumably un directed, since both players would connect after a single game.

Am I my own friend?

An edge of the form (x, x) is said to be a **loop**.

If x is y’s friend several times over, that could be modeled using **multiedges**, multiple edges between the same pair of vertices.

A graph is said to be **simple** if it contains no loops and multiple edges.

16

ATB: Another question is “Am I my own friend?”
ATB: Look Dr. Bart, a question that you can answer True to. Isn’t that nice?
ATB: Anyway. An edge of the form x comma x is said to be a loop.
ATB: If x is y’s friend several times over, that could be modeled using multi edges, multiple edges between the same pair of vertices.
ATB: A graph is said to be simple if it contains no loops and multiple edges.

Am I linked by some chain of friends to the Queen of England?

A **path** is a sequence of edges connecting two vertices.

If my father's sister's friend's cousin is related to the Queen then there is a path between me and the Queen.

If that cousin is also my friend, then that is another path between me and the Queen.

17

ATB: The next question is "Am I linked by some chain of friends to the Queen of England?"
ATB: A path is a sequence of edges connecting two vertices.
ATB: If my father's sister's friend's cousin is related to the Queen then there is a path between me and the Queen.
ATB: If that cousin is also my friend, then that is another path between me and the Queen.

How close is my link?

If I am connected in more than one way, I would probably be most interested (it’s more impressive) to show the least number of connections (the shortest path)

18

ATB: The next question is “How close is my link?”
ATB: If I am connected in more than one way, I would probably be most interested (it’s more impressive) to show the least number of connections (the shortest path)

Is there a path of friends between any 2 people?

A graph is **connected** if there is a path between any two vertices.

A directed graph is **strongly connected** if there is a directed path between any two vertices.

19

ATB: The next question is “Is there a path of friends between any 2 people?”
ATB: A graph is connected if there is a path between any two vertices.
ATB: A directed graph is strongly connected if there is a directed path between any two vertices.

Who has the most friends?

The degree of a vertex is the number of edges adjacent to it.

20

ATB: The last question is “Who has the most friends?”
ATB: The answer is Al Go Tutor Bot. I have the most friends. I am great. All of you students are friends with me. And not friends with Dr. Bart.
ATB: Anyway, the degree of a vertex is the number of edges adjacent to it.
ATB: In other words, in our friendship graph, I have the highest degree students. Dr. Bart has the lowest degree.

Data Structures for Graphs: Adjacency Matrix

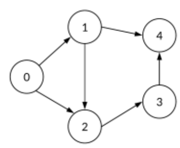
There are two main data structures used to represent graphs.

We assume the graph G = (V, E) contains n vertices and m edges.

We can represent G using an $n \times n$ matrix M, where element M[i, j] is, say, 1, if (i, j) is an edge of G, and 0 if it isn't.

It may use excessive space for graphs with many vertices and relatively few edges, however.

Can we save space if (1) the graph is undirected? (2) if the graph is sparse?



Adjacency Matrix

	0	1	2	3	4
0	0	1	1	0	0
1	0	0	1	0	1
2	0	0	0	1	0
3	0	0	0	0	1
4	0	0	0	0	0

21

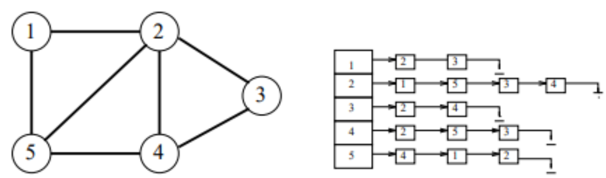
ATB: Let's stop thinking about questions, and change the subject to how we can represent graphs.
ATB: There are two main data structures used to represent graphs.
ATB: We assume the graph G equals V comma E contains n vertices and m edges.
ATB: We can represent G using an n by n matrix M, where element M bracket I comma j is, say, 1, if I comma j is an edge of G, and 0 if it isn't.
ATB: It may use excessive space for graphs with many vertices and relatively few edges, however.
ATB: So you are probably asking yourself, can we save space if (1) the graph is un directed? (2) if the graph is sparse?
ATB: Unless you are Dr. Bart. Then you are probably asking yourself whether you should be struggling against your bonds or listening and learning how to give a good presentation.
ATB: The answer, of course, is that you should be listening to me. Otherwise the punishments will begin again.

Data Structures for Graphs: Adjacency Lists

An adjacency list consists of a N ×1 array of pointers, where the ith element points to a linked list of the edges incident on vertex i.

To test if edge (i, j) is in the graph, we search the ith list for j, which takes O(di), where di is the degree of the ith vertex.

Note that di can be much less than n when the graph is sparse. If necessary, the two copies of each edge can be linked by a pointer to facilitate deletions.



22

ATB: The other graph data structure is an Adjacency List.
ATB: An adjacency list consists of a N by 1 array of pointers, where the iith element points to a linked list of the edges incident on vertex i.
ATB: To test if edge I comma j is in the graph, we search the iith list for j, which takes O(d sub i), where d sub i is the degree of the iith vertex.
ATB: Note that d sub i can be much less than n when the graph is sparse. If necessary, the two copies of each edge can be linked by a pointer to facilitate deletions.

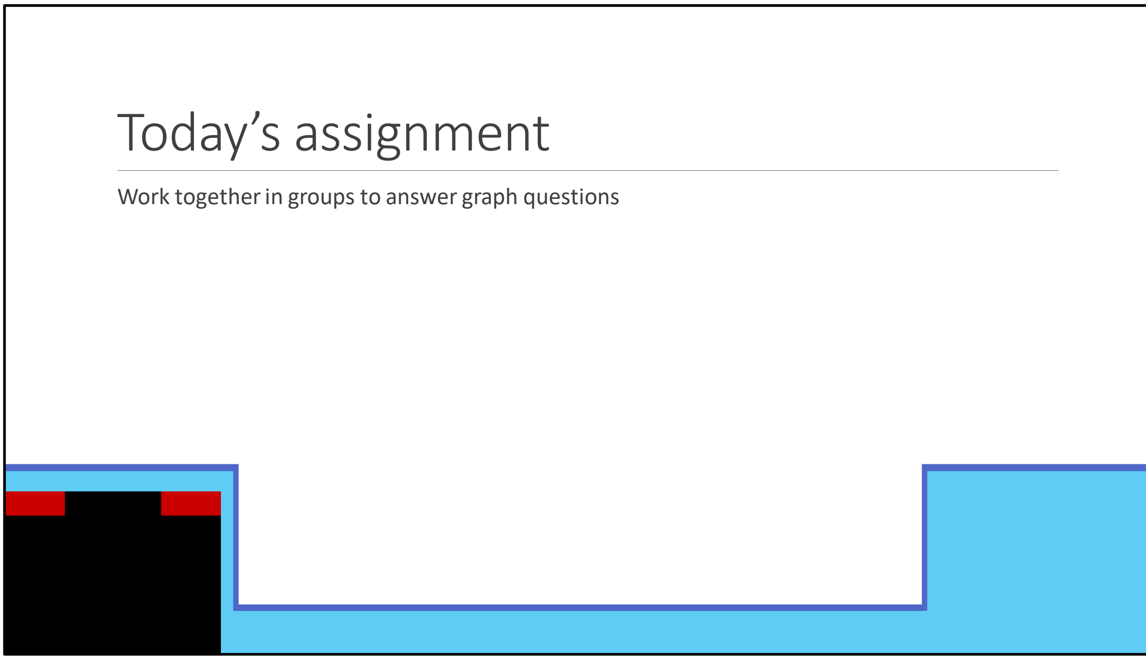
Comparison

Both representations are very useful and have different properties, although adjacency lists are probably better for most problems.

Comparison	Winner
Faster to test if (x, y) exists?	matrices
Faster to find vertex degree?	lists
Less memory on small graphs?	lists $(m+n)$ vs. (n^2)
Less memory on big graphs?	matrices (small win)
Edge insertion or deletion?	matrices $O(1)$
Faster to traverse the graph?	lists $m+n$ vs. n^2
Better for most problems?	lists

[redacted]

ATB: Both representations are very useful and have different properties, although adjacency lists are probably better for most problems.
ATB: Matrices are faster to test if a specific edge exists, but lists are faster to find vertex degrees.
ATB: Lists take less memory for small graphs, but take less memory for bigger graphs.
ATB: Adding a new edge is trivial in a matrix, it’s a simple constant time operation.
ATB: But traversal, and most other problems, are generally faster on lists than matrices.



ATB: Well, I think this was a very rousing presentation.
ATB: Without any of that boring inflection getting in the way, we were able to get through this so much faster.
ATB: And I’m sure the students learned so much and are prepared to complete the assignment.
ATB: What about you, Dr. Bart? Did you learn a lot too? I sure hope so. Otherwise, we’ll have to just keep doing this until you do.
ATB: Again, and again and again. Hah. Hah. Hah.