Concepts of Computation

Session 9b
Graph Algorithms

Dr Oded Lachish
Email: oded@dcs.bbk.ac.uk

(Slides prepared with the support of Dr Paul Newman and Eva Szatmari)
Graph Algorithms

• Breadth First Search (BFS)
• Checking if a graph is bipartite
• Graph theoretic problems for general knowledge
• Hamilton path and cycle
• \( K \)-Clique
Graphs and computers

How does a computer access a graph \( G=(V,E) \)?

One of the ways is:

The computer has access to the set of vertices \( V=\{v_1, v_2, \ldots, v_n\} \) \( \{v_1, v_2, \ldots, v_n\} \) means \( v_i \) for every \( i \) from 1 to \( n \), where \( n \) is the size of the set \( V \)

The computer also has access to the set \( E \) in the following form:

- It can check for any vertex of its choice in \( V \), what are the vertices adjacent to it.

In the graph on the right the vertices

\( v_3, v_5, v_7, v_8 \) are adjacent to \( v_4 \)

Another way to say this is that

\( v_3, v_5, v_7, v_8 \) are neighbours of \( v_4 \)
Graphs and computers

What can we do with what we learned in the previous slide
We can get the computer to find the distance between $v_1$ and $v_8$

How? (remember the computer doesn’t see the graph like we do)

1. Let $V_0 = \{v_1\}$
2. Find all the neighbours of $v_0$ and put them in a set $V_1$. ($V_1 = \{v_2, v_6\}$)
3. For every vertex in $V_1$, find all its neighbours and if they are not in $V_0 \cup V_1$, then add them to $V_2$. ($V_2 = \{v_3, v_5\}$)
4. For every vertex in $V_2$, find all its neighbours and if they are not in $V_0 \cup V_1 \cup V_2$, then add them to $V_3$. ($V_3 = \{v_4\}$)
5. In the same way we get $V_4 = \{v_7, v_8\}$ and
6. $V_5 = {}$ this means we are done.

What next?
Graphs and computers

We got

\( V_0 = \{v_1\}, \ V_1 = \{v_2, v_6\}, \ V_2 = \{v_3, v_5\}, \ V_3 = \{v_4\}, \ V_4 = \{v_7, v_8\} \) and \( V_5 = \{\}\)

This means that

\[ d(v_1, v_1) = 0 \]
\[ d(v_1, v_2) = 1, \ d(v_1, v_6) = 1, \]
\[ d(v_1, v_3) = 2, \ d(v_1, v_5) = 2 \]
\[ d(v_1, v_4) = 3 \]
\[ d(v_1, v_7) = 4, \ d(v_1, v_8) = 4 \]

We got the answer to the question we asked and more.

This algorithm is called Breadth First Search.
Breadth First Search (BFS)

How do we do BFS in general:
1. Pick a vertex of our choice and put it in $V_0$
2. Set $i$ to 0
3. do
   • If $V_i$ is empty stop
   • If $V_i$ is not empty, then for every vertex in $V_i$, find all its neighbours and if they are not in a set $V_j$ such that $j \leq i$ and add them to $V_{i+1}$.
   • Increase $i$ by 1
4. Repeat step 3

The sets the computer found tell the distance from the vertex you picked to every other vertex in the graph!
Bipartite graphs

Recall, in a bipartite graph $G$, the vertex set $V$ is partitioned into two subsets (say $S$ and $T$).

What if someone tells us a graph is bipartite but does not provide us with the sets?

Is the following graphs bipartite?
Finding sets for a Bipartite graph

Is the following graphs bipartite?

We can take the BFS and change it to get an algorithm that solves the problem:

1. Pick a vertex of our choice and put it in $V_0$
2. Set $i$ to 0 and $V_1$ to be the empty set
3. do
   • If $V_i$ is not empty, then for every vertex in $V_i$, find all its neighbours and add them to $V_{1-i}$.
   • If $V_{1-i}$ did not change, then stop
   • If $i$ is 1, then set it to 0, and if $i$ is 0, then set it to 1
4. Repeat step 3

This works for graphs that are connected, if they aren’t connected. It is done for every connected component separately.

In the next slide we shall see how this works.
Checking if a graph is Bipartite

1. \( V_0 = \{v_3\} \)
2. The vertices adjacent to \( v_3 \) are \( v_1, v_2 \) and \( v_8 \) so we add the to \( V_{1-0} = V_1 \) and hence now \( V_1 = \{v_1, v_2, v_8\} \)
3. The vertices adjacent to the vertices in \( V_1 \) are \( v_3, v_4, v_5 \) and \( v_7 \) and hence now \( V_0 = \{v_3, v_4, v_5, v_7\} \)
4. The vertices adjacent to the vertices in \( V_0 \) are \( v_1, v_2, v_6 \) and \( v_8 \) and hence now \( V_1 = \{v_1, v_2, v_6, v_8\} \)
5. The vertices adjacent to the vertices in \( V_1 \) are \( v_3, v_4, v_5 \) and \( v_7 \) and hence now \( V_0 = \{v_3, v_4, v_5, v_7\} \)
6. \( V_0 \) hasn’t changed so we are done
Checking if a graph is Bipartite

The graph following graph is bipartite
With sets \( \{v_1, v_2, v_6, v_8\} \) and \( \{v_3, v_4, v_5, v_7\} \)

We can draw it in the manner we learned for bipartite graphs
What if the graph is not bipartite?

If this happens, then for at least one of $V_0$ and $V_1$ there exists an edge between two of its vertices.

So when we get the sets $V_0$ and $V_1$ from the algorithm we have to check this. If it happened then the graph is not bipartite.

Is the another way to know that a graph is not bipartite?
All the closed walks in Bipartite graphs have an even length

If we follow a walk on a bipartite graph then every edge moves us to the other side

For example: \( \{v_1, v_4, v_2, v_3, v_1\} \)

So we go forth and back repeatedly until we return to where we started, each time we did that counts for 2 edges

So the total length must be a multiple of 2
All the closed walks in Bipartite graphs have an even length

Conclusion:

a graph is bipartite if and only if all its closed walks have even length

The graph below is not bipartite, because it has a cycle of length 3 (which is a closed walk of length 3) \((v_8, v_6, v_5)\)

Note: obviously to check if a graph is bipartite you don’t need to check every closed walk (you can’t), Instead you use the algorithm we

We got this graph by adding an edge to a graph from a previous example
A graph is bipartite if and only if all its closed walks have even length.

To check if a graph is bipartite, you don’t need to check every closed walk (you can’t), instead you use the algorithm.

The advantage of the closed walk with odd length is:

- If someone provides you with a closed walk with an odd length, then in order to believe that the graph is not bipartite,
  - you only need to check that this is indeed a walk, which a lot less than to run the algorithm we learned.
Hamilton path,

A Hamiltonian path is a path in which every vertex participates (recall that the fact that it is a path means that every vertex must participate exactly once.

\((v_1, v_2, v_3, v_9, v_7, v_8, v_4, v_5, v_6)\) — in the above example this is a Hamilton path

A graph may have a number of different Hamilton Paths.
Hamilton path, cycle

A Hamiltonian cycle is a cycle in which every vertex participates (recall that the fact that it is a cycle means that every vertex must participate exactly once, excluding the first).

\((v_1, v_2, v_3, v_9, v_7, v_8, v_4, v_5, v_6, v_1)\) — in the above example this is a Hamilton cycle

A graph may have a number of different Hamilton Cycles.
Hamilton path and cycle

• The graph below has a Hamiltonian path but not a Hamiltonian Cycle

Can you find the path?
Can you explain why it does not have a cycle?
Hamilton path and cycle

• Answers

• The path is on the slide

• There is no cycle because,
  • if it started at \( v_1 \)
  • it has to go through \( v_4 \) on the way to \( v_7 \) and then again on the way back to \( v_1 \)
  • If there is a Hamiltonian cycle starting at any other vertex, then there is a Hamiltonian path starting at \( v_1 \)
Hamilton path, cycle

How to solve a question about this in the exam:
If asked you should be able to find it?
The answer to why a graph doesn’t have a Hamiltonian Path or Cycle you should be able to give an explanation like we just gave (why a specific vertex must be used twice).
**k-clique**

- A clique is a complete subgraph.
- In the *k*-clique problem the goal is to find a subgraph \(H\) of a graph \(G\), so that \(H\) has *k* vertices and is a complete graph.

![Graph with vertices and edges](image)

- In the graph above the induced subgraph with vertex set \(\{v_2, v_3, v_5, v_6\}\) is a clique of size 4 since it has all the possible edges. So, the graph in the example has a 4-clique!
**k-clique**

- In the graph above the induced subgraph with vertex set \{v_2, v_3, v_5, v_6\} is a clique of size 4 since it has all the possible edges. So, the graph in the example has a 4-clique!

Does the graph have a 5-clique?

No,

In a 5-clique every vertex has degree 4, so a graph that has a 5-clique must have at least 5 vertices of degree 4.

The graph above has 5 vertices with degree 4 \{v_2, v_3, v_5, v_6, v_4\}, but they don’t form a clique, because there is no edge including \(v_6\) and \(v_4\), so it does not have a 5-clique!
**$k$-clique**

- Does having 5 vertices with degree 4 mean that the graph has a 5-clique? No,
The graph in the above example has 9 vertices with degree at least 4, but does not have a 5-clique.

Is there an easy way to find a $k$-clique?
Probably not, we don’t know one.
Currently, the best we can do is running time that is not far better than $n^k$. This is bad, if $n=50$ and $k=20$, the number is already too big.
**k-clique**

How to solve a question about this in the exam:
If asked you should be able to find it?
The answer to why a graph doesn’t have a clique of size more than say 4 is based on the degree explanation we just gave.