Implicit Representation of Graphs

Md. Saidur Rahman

Department of Computer Science and Engineering,
Bangladesh University of Engineering and Technology, Dhaka

October 2012

Outline

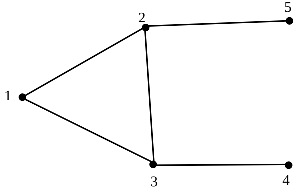
Implicit Representation of Graphs

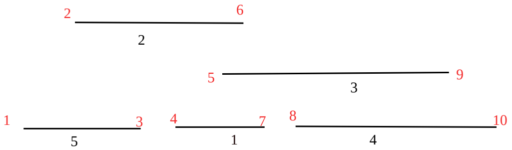
R is an implicit representation of a graph G if R assigns $O(logn)$ bits to each vertex of G, and there is an adjacency testing algorithm which decides whether x and y are adjacent based only on the bits stored at x and y.

Implicit Representation of Graphs

R is an implicit representation of a graph G if R assigns $O(logn)$ bits to each vertex of G, and there is an adjacency testing algorithm which decides whether x and y are adjacent based only on the bits stored at x and y.

An Implicit Representation of an Interval Graph.





1: 4, 7
2: 2, 6
3: 5, 9
4: 8,10
5: 1, 3

Space optimal Representation and Implicit Representation

Is the implicit representation of an interval graph space optimal?

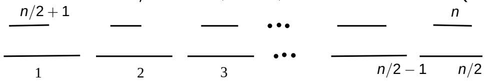
- The number of labeled interval graphs is $2^{\Omega(n \log n)}$.

Space optimal Representation and Implicit Representation

Is the implicit representation of an interval graph space optimal?

- The number of labeled interval graphs is $2^{\Omega(n \log n)}$.

Proof. Consider the following interval graph. Each of the vertices in the set $1, \cdots, n/2$ is matched with exactly one vertex in the set $n/2+1, \cdots, n$. There are $(n/2)!$ such interval graphs.

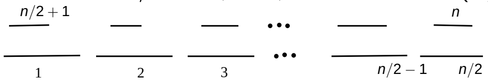


Space optimal Representation and Implicit Representation

Is the implicit representation of an interval graph space optimal?

- The number of labeled interval graphs is $2^{\Omega(nlogn)}$.

Proof. Consider the following interval graph. Each of the vertices in the set $1, \cdots, n/2$ is matched with exactly one vertex in the set $n/2+1, \cdots, n$. There are $(n/2)!$ such interval graphs.



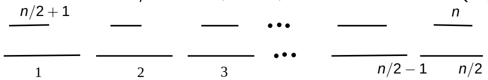
$$(n/2)! = (n/2)(n/2-1)\cdots(n/4)(n/4-1)\cdots 1 > (n/2)(n/2-1)\cdots(n/4+1)(n/4) > (n/4)^{n/4} = 2^{(n/4)log(n/4)} = 2^{\Omega(nlogn)}$$

Space optimal Representation and Implicit Representation

Is the implicit representation of an interval graph space optimal?

- The number of labeled interval graphs is $2^{\Omega(nlogn)}$.

Proof. Consider the following interval graph. Each of the vertices in the set $1, \cdots, n/2$ is matched with exactly one vertex in the set $n/2+1, \cdots, n$. There are $(n/2)!$ such interval graphs.



$(n/2)! = (n/2)(n/2-1)\cdots(n/4)(n/4-1)\cdots 1 > (n/2)(n/2-1)\cdots(n/4+1)(n/4) > (n/4)^{n/4} = 2^{(n/4)log(n/4)} = 2^{\Omega(nlogn)}$

Representation of all interval graphs requires $\Omega(nlogn)$ bits.

We achieve $O(nlogn)$ bits with the implicit representation.

Thus implicit representation is space optimal.

Implicit Representation

Does every graph class admit implicit representation?

Implicit Representation

Does every graph class admit implicit representation?
No.

Implicit Representation

Does every graph class admit implicit representation?
No.

- There are classes of graphs which have $2^{\Theta(n^2)}$ members. Any representation of graphs of such classes must use $\Omega(n^2)$ bits for the average graphs in the class, while an implicit representation uses $O(nlogn)$ bits.

Implicit Representation

Does every graph class admit implicit representation?
No.

- There are classes of graphs which have $2^{\Theta(n^2)}$ members. Any representation of graphs of such classes must use $\Omega(n^2)$ bits for the average graphs in the class, while an implicit representation uses $O(nlogn)$ bits.
- There are graph classes with $2^{O(nlogn)}$ members which do not have implicit representation.

Implicit Representation

Does every graph class admit implicit representation?
No.

- There are classes of graphs which have $2^{\Theta(n^2)}$ members. Any representation of graphs of such classes must use $\Omega(n^2)$ bits for the average graphs in the class, while an implicit representation uses $O(nlogn)$ bits.
- There are graph classes with $2^{O(nlogn)}$ members which do not have implicit representation. Example: 7-sparse graph.

Outline

Implicit Representation of Trees

A graph of n vertices is a *7-sparse graph* if it contains at most $7n$ edges.

Claim: The 7-sparse graphs do not have an implicit representation.

Proof. Suppose that the 7-sparse graphs have an implicit representation. Consider the set of all graphs on n vertices; we know that there are $2^{\Theta(n^2)}$ such graphs. Let G be an arbitrary graph of n vertices. Add n^2 isolated vertices to G, and we have a graph G' which is 7-sparse. G' must have an implicit representation. Adjacencies between vertices in an implicit representation can be determined using only the bits stored at the vertices, so G can be reconstructed from the bits stored at each vertex of G within the representation of G'.

Implicit Representation of Trees

There are $O(log(n^2 + n)$ bits stored at each vertex of G'; thus G can be reconstructed from a set of $O(nlogn)$ bits. This implies that there are $2^{O(nlogn)}$ graphs on n vertices, which is certainly false. So there can not be an implicit representation of the 7-sparse graphs.

Outline

Implicit Representation of Trees

- An adjacency list representation of a tree uses $O(nlogn)$ bits.

Implicit Representation of Trees

- An adjacency list representation of a tree uses $O(nlogn)$ bits.
- Therefore, there are $2^{O(nlogn)}$ trees of n vertices.

Implicit Representation of Trees

- An adjacency list representation of a tree uses $O(nlogn)$ bits.
- Therefore, there are $2^{O(nlogn)}$ trees of n vertices.
- But adjacency list representation does not imply implicit representation, since a tree can have vertices of arbitrarily high degree.

Implicit Representation of Trees

A simple Implicit representation of a tree:
Make the tree a rooted tree by choosing an arbitrary vertex as the root.

Implicit Representation of Trees

A simple Implicit representation of a tree:
Make the tree a rooted tree by choosing an arbitrary vertex as the root.
Store only the parent of each vertex v at v.

Implicit Representation of Trees

A simple Implicit representation of a tree:
Make the tree a rooted tree by choosing an arbitrary vertex as the root.
Store only the parent of each vertex v at v.
Two vertices v and w are adjacent if and only if v is the parent of w or w is the parent of v.

Outline

Implicit Representation of Planar Graphs

Every planar graph has a vertex of degree at most 5.

Implicit Representation of Planar Graphs

Every planar graph has a vertex of degree at most 5.
Simple implicit representaton:

Implicit Representation of Planar Graphs

Every planar graph has a vertex of degree at most 5.
Simple implicit representaton:
Repeatedly remove vertices of degree at most 5 and store all remaining neighbors of the vertex with the vertex.

Outline

Hereditary Graphs

A class of graphs is heriditary if for every graph G in the class, every induced subgraph of G is also in the class.

Hereditary Graphs

A class of graphs is heriditary if for every graph G in the class, every induced subgraph of G is also in the class.
Examples: Interval Graphs, Planar graphs, triangle free graphs etc; deleting vertices will not disturb the property.

Hereditary Graphs

A class of graphs is heriditary if for every graph G in the class, every induced subgraph of G is also in the class.
Examples: Interval Graphs, Planar graphs, triangle free graphs etc; deleting vertices will not disturb the property.
Connected graphs are not heditary class.

Hereditary Graphs

A class of graphs is heriditary if for every graph G in the class, every induced subgraph of G is also in the class.
Examples: Interval Graphs, Planar graphs, triangle free graphs etc; deleting vertices will not disturb the property.
Connected graphs are not heditary class.
Trees are not hereditary class.

Hereditary Graphs

A class of graphs is heriditary if for every graph G in the class, every induced subgraph of G is also in the class.
Examples: Interval Graphs, Planar graphs, triangle free graphs etc; deleting vertices will not disturb the property.
Connected graphs are not heditary class.
Trees are not hereditary class.
Forests are a hereditary class.

Hereditary Graphs

A class of graphs is heriditary if for every graph G in the class, every induced subgraph of G is also in the class.

Hereditary Graphs

A class of graphs is heriditary if for every graph G in the class, every induced subgraph of G is also in the class.
Examples: Interval Graphs, Planar graphs, triangle free graphs etc; deleting vertices will not disturb the property.

Hereditary Graphs

A class of graphs is heriditary if for every graph G in the class, every induced subgraph of G is also in the class.
Examples: Interval Graphs, Planar graphs, triangle free graphs etc; deleting vertices will not disturb the property.
Connected graphs are not heditary class.

Hereditary Graphs

A class of graphs is heriditary if for every graph G in the class, every induced subgraph of G is also in the class.
Examples: Interval Graphs, Planar graphs, triangle free graphs etc; deleting vertices will not disturb the property.
Connected graphs are not heditary class.
Trees are not hereditary class.

Hereditary Graphs

A class of graphs is heriditary if for every graph G in the class, every induced subgraph of G is also in the class.
Examples: Interval Graphs, Planar graphs, triangle free graphs etc; deleting vertices will not disturb the property.
Connected graphs are not heditary class.
Trees are not hereditary class.
Forests are a hereditary class.

Hereditary Graphs

- Any heriditary class of graphs with at most cn edges has an implicit representations.

Hereditary Graphs

- Any heridітary class of graphs with at most cn edges has an implicit representations.

Proof

Suppose a hereditary graph class have at most cn edges. Then the total size of the adjacency list is $2cn$. This implies that there must be some vertex with degree atmost $2c$ neighbors. We remove this vertex and store all neighbors with this vertex.

Hereditary Graphs

- Any heriditary class of graphs with at most cn edges has an implicit representations.

Proof

Suppose a hereditary graph class have at most cn edges. Then the total size of the adjacency list is $2cn$. This implies that there must be some vertex with degree atmost $2c$ neighbors. We remove this vertex and store all neighbors with this vertex. Since the class is hereditary, the remaining graph has atmost $c(n-1)$ edges, the total size of adjacency lists is at most $2c(n-1)$, and there is again a neighbor with at most $2c$ neighbors. Repeting the process we get an implicit representation with at most $2c$ integers stored at each vertex.

Hereditary Graphs

Reference: Jeremy P. Spinard, Efficient Graph Representations, American Mathematical Society, 2003.