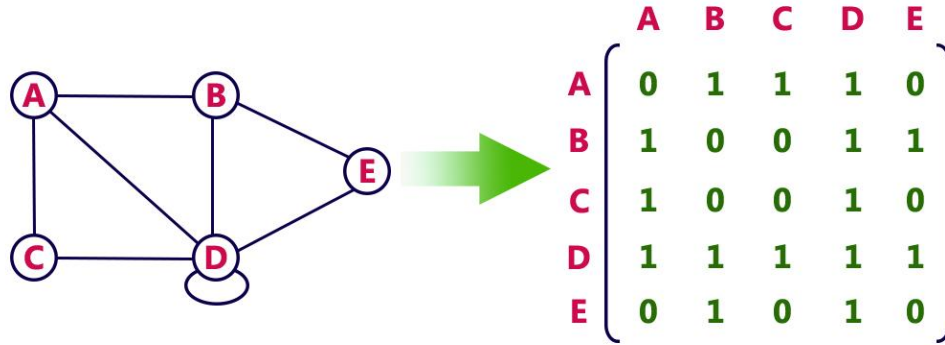


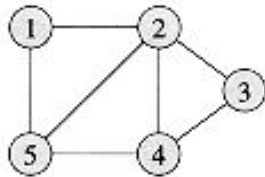
Converting A Graph Represented As An Adjacency List Into An Adjacency Matrix Using Multithreading



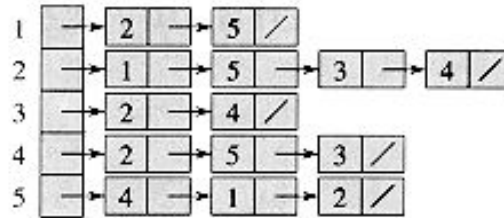
By: Maurely Acosta, Yair Camacho, Jorge Costa, and Malachi Williams.

Introduction

- **Language Used:** C
- **Main Idea:**
 - Compare **3** approaches
 - Sequential Linear, Multithreaded Linear, and Brute Force.
- **What is an adjacency list?**
 - Each head node contains the edges that a node is adjacent to.
- **What is an adjacency matrix?**
 - If an element in the matrix has a 1, it means the nodes are adjacent. If 0, they are not adjacent



(a)



(b)

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

(c)

Process

- Brute force approach
 - Checks every potential adjacency
 - Uses multithreading
- Sequential linear approach
 - Checks adjacencies for each vertex
- Multithreaded linear approach
 - Also checks adjacencies for each vertex
 - Uses multithreading

Brute Force Approach

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



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

Brute Force Approach Psuedocode

```
for(row = 0; row < graph.size; row++)  
{  
    for(column = 0; column < graph.size; column++)  
    {  
        if( vertex(row) is adjacent to vertex(column) ) {matrix[row][column] = 1};  
    }  
}
```

Linear Approach

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



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

Linear Approach Pseudocode

```
for(row = 0; row < graph.size; row++)
```

```
{
```

```
    node next = adjList[row]
```

```
    while( next != NULL)
```

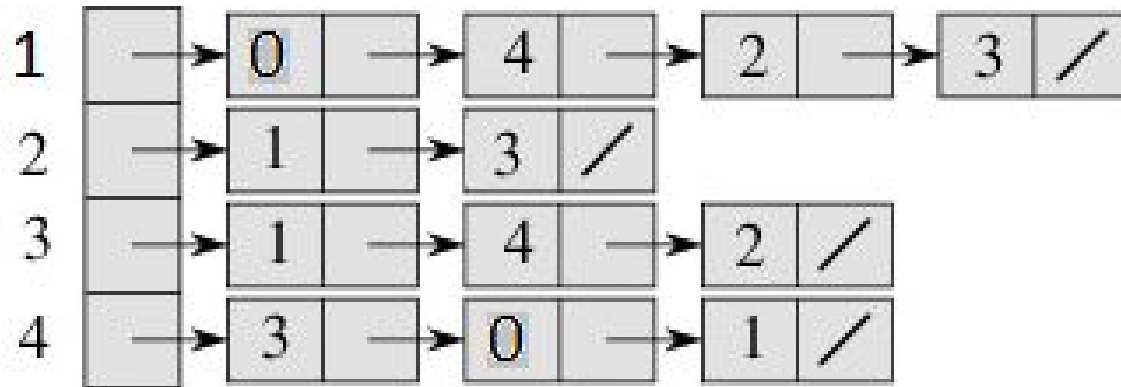
```
    {
```

```
        Matrix[row][next.column] = 1;
```

```
    }
```

```
}
```

Linear Approach



Results

Time (milliseconds)

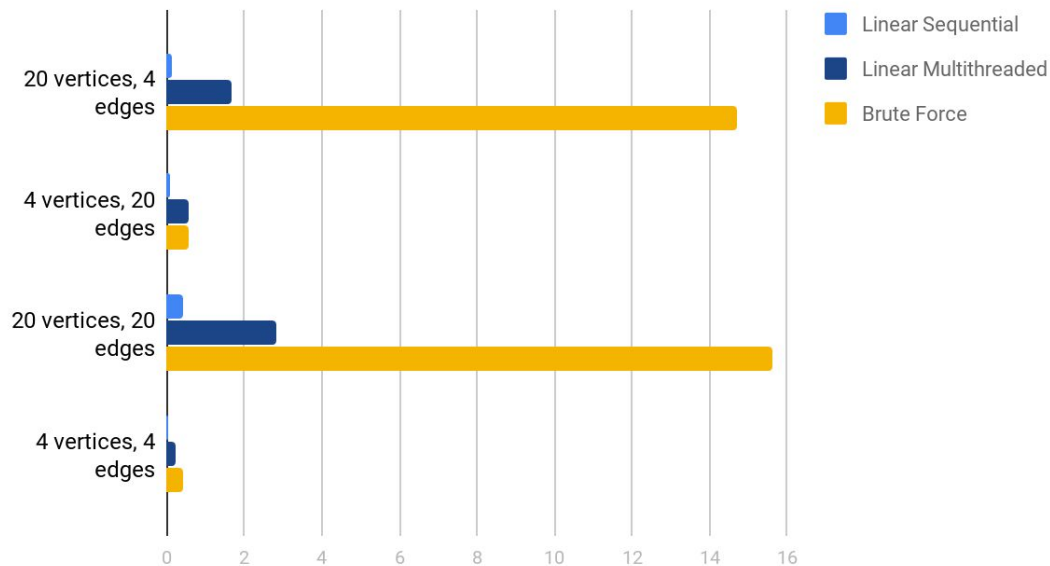


Table 1

	Linear Sequential	Linear Multithreaded	Brute Force
20 vertices, each has 4 edges	0.148 ms	1.69 ms	14.68 ms
4 vertices, each has 20 edges	0.098 ms	0.58 ms	0.57 ms
20 vertices, each has 20 edges	0.43 ms	2.82 ms	15.60 ms
4 vertices, each has 4 edges	0.025 ms	0.25 ms	0.44 ms

Questions

