CSCI2100C Lab 4

Prepared by:

Song BIAN

Reminders

- Please remember your password for the online judge
- Please start your assignment early
  - And report any issues as early as possible. Some issues regarding the registration of the OJ were reported not until some hours before the deadline. For such cases of failure of code submission, we may not grade your lab.
  - Penalty:
    - -10 marks/day pro rata for first two days after deadline
    - -10 marks/hour pro rata afterwards (so you get 0 marks if you submit 2 day 8 hours after the deadline)
- Grading is based on the **last** submission
- Write your own code
  - We will check your code
  - Suspected cases of plagiarism will be reported
- Questions?

Agenda

- BFS with AdjMatrix
- Double Hashing
- Overview of Lab 4 Problems

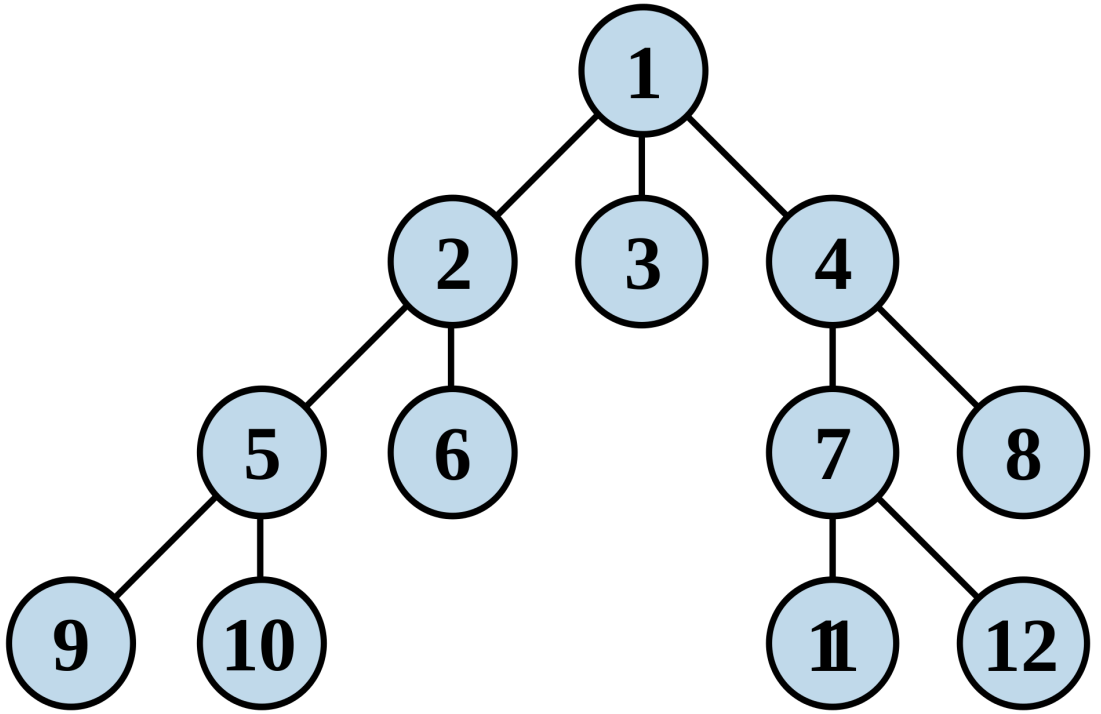
Agenda

- **BFS with AdjMatrix**
- Double Hashing
- Overview of Lab 4 Problems

BFS with AdjMatrix

Properties

This non-recursive implementation is similar to the non-recursive implementation of depth-first search, but differs from it in two ways: it uses a queue (First In First Out) instead of a stack and it checks whether a vertex has been discovered before enqueueing the vertex rather than delaying this check until the vertex is dequeued from the queue.
The *Q* queue contains the frontier along which the algorithm is currently searching.
Nodes can be labelled as discovered by storing them in a set, or by an attribute on each node, depending on the implementation.



Agenda

- BFS with AdjMatrix
- **Double Hashing**
- Overview of Lab 4 Problems

Double Hashing

Properties

- Double hashing
  - We have an additional hash function $h' > 0$.
  - Insertion: we probe $h(k,i) = \big(h(k) + i \cdot h'(k)\big)\%m$ one by one for i from 0 to $m-1$ until an empty slot is found.
  - Search: we search $h(k,i)$ for i from 0 to $m-1$ until one of the following happens:
    - $T[h(k,i)]$ has the record with key equal to k.
    - $T[h(k,i)]$ is empty, then no record contains key k in the hash table

Agenda

- BFS with AdjMatrix
- Double Hashing
- **Overview of Lab 4 Problems**

Lab 4

Problem 1

- Straight forward implementation of BST with AdjMatrix

Code segment to complete:

```
void bfs(struct Graph * G, int S){
    // write your code here
    // use the provided printVertex function for output of a vertex
}
```

BST with AdjMatrix

Description:

Given an unweighted, undirected and connected graph and a vertex, find out the BFS traversal order of the graph from the vertex. When there are multiple neighbours for a vertex, visit the neighbours in ascending order of their vertices' label. Note that the required traversal is unique.

Input:

The first line are two integers, the number of vertices N and edges E.

The second line is one integer, the starting vertex S.

The next E lines are the edges, each consist of two integers, u and v, representing an undirected edge between vertex u and v.

Note that the vertices are labeled from 0 to N-1.

2 <= N <= 1,000; 0 <= E <= 100,000

Output:

Output the BFS traversal from S as required.

Lab 4

Problem 2

- Straight forward implementation of Double Hashing

Code segment to complete:

```
int search(int x, struct Entry hashTable[], int hashTableSize){
    // write your code here
}

void insert(int x, struct Entry hashTable[], int hashTableSize){
    for (int i = 0; i < hashTableSize - 1; i++){
        int hashed_key = doubleHashFunc(x,i,hashTableSize);
        if(hashTable[hashed_key].ID == -1){
            // write your code here
        }
    }
}
```

Double Hashing

In a museum, there is a infra-red sensor at the door scanning the the ID of visitors, which is recorded each time a visitor enters or leaves the museum. Given a series of IDs recorded in a day, your task is to determine whether every visitor has left the museum. Assume nobody is inside the museum at the beginning.

Input:

N lines of ID: M_1, M_2, ..., M_N.

0 <= N <= 10000; 1 <= M_i <= 2^31-1;

Finally, input -1 as the end symbol.

Output:

If everyone has left, output -1. Otherwise, output the number of the people that are still inside the museum.

Sample Input 1:

2

5

2

7

9

-1

Sample Output 1:

3

Last but not Least

- Please add this declaration on top of (commented as shown) all your codes submitted to the OJ.

```
/*
I, <Your Full Name>, am submitting the assignment for
an individual project.
I declare that the assignment here submitted is original except for
source material explicitly acknowledged, the piece of work, or a
part
of the piece of work has not been submitted for more than one
purpose
(i.e. to satisfy the requirements in two different courses) without
declaration. I also acknowledge that I am aware of University
policy
and regulations on honesty in academic work, and of the
disciplinary
guidelines and procedures applicable to breaches of such policy and
regulations, as contained in the University website
http://www.cuhk.edu.hk/policy/academichonesty/.
It is also understood that assignments without a properly signed
declaration by the student concerned will not be graded by the
teacher(s).
*/
```

Questions?