# CS403 Parallel Programming Mini Project 1 - Graph Search

Rounak Jangir Student ID : 201352028

Yash Choubey Student ID: 201351006

Anjul Kumar Tyagi Student ID : 201351033

# 1 Graph Search Techniques

### 1.1 Breadth First Search

#### 1.1.1 Pseudocode

**Input:** A starting vertex (s) and total number of vertices (n). **Output:** All vertices reachable from root labeled as explored.

```
BFS(s,n)
  Set all nodes to "not visited";
  q.enqueue(initial node);
  vis[s]=1//set it as visited
  p=q.dequeue();
  if(p is not zero)
     print p;
  while(p is not zero)
     for(i=1 to n)
         if(a[p][i] is not equal to 0 and node is not visited)
         {
            q.enqueue(i) //node i
            vis[i]=1 //set it as visited
     p=q.dequeue();
     if(p is not equal to 0)
       print p;
  }
  for(i=1 to n)
     if (i is not visited)
       BFS(i,n)
}
```

# 1.2 Depth First search

## 1.2.1 Pseudocode

**Input:** A starting vertex (s) and total number of vertices (n). **Output:** All vertices reachable from root labeled as explored.

```
DFS(s,n)
  push(s);
  vis[s]=1 //set it as visited
  k=pop();
  if(k not equal to 0)
     print k;
  while(k not equal to 0)
     for(i=1 to n)
        if(a[k][i]not equal to 0 and node is not visited)
        {
          push(i);
          vis[i]=1;// set it as visited
        }
     k=pop();
     if(k not equal to 0)
        print k;
  }
  for(i=1 to n)
     if(node is not visited)
  dfs(i,n);
}
```

## 1.3 Analysis of Outputs

• Below figure shows the output of the cache miss rate.

```
==5968==
==5968== I
                         278,558
             refs:
==5968== I1 misses:
                             964
==5968== LLi misses:
                             954
==5968== I1 miss rate:
                            0.34%
==5968== LLi miss rate:
                            0.34%
==5968==
==5968== D
             refs:
                         113,078
                                   (72,356 rd
                                                + 40,722 wr)
==5968== D1
             misses:
                           1,795
                                   ( 1,275 rd
                                                      520 Wr)
                                    1,086 rd
                           1,566
                                                      480 Wr)
 =5968== LLd misses:
==5968== D1 miss rate:
                             1.5%
                                       1.7%
                                                      1.2%
==5968== LLd miss rate:
                             1.3% (
                                       1.5%
                                                      1.1%
==5968==
==5968== LL refs:
                           2,759
                                   ( 2,239 rd
                                                      520 Wr)
==5968== LL misses:
                           2,520
                                                      480 Wr)
                                    2,040 rd
                                      0.5%
==5968== LL miss rate:
                             0.6%
                                                      1.1%
```

Figure 1: Cache misses

• Given below is the figure showing the memory leaks analysis.

```
==5924== Memcheck, a memory error detector
==5924== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al.
==5924== Using Valgrind-3.10.1 and LibVEX; rerun with -h for copyright info
=5924== Command: ./serial
==5924==
=5924==
==5924== HEAP SUMMARY:
==5924==
            in use at exit: 0 bytes in 0 blocks
==5924==
          total heap usage: 0 allocs, 0 frees, 0 bytes allocated
==5924==
=5924== All heap blocks were freed -- no leaks are possible
==5924==
 =5924== For counts of detected and suppressed errors, rerun with: -v
=5924== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

Figure 2: Memory leaks

• Below figure shows the contribution of each of the functions from the given code.

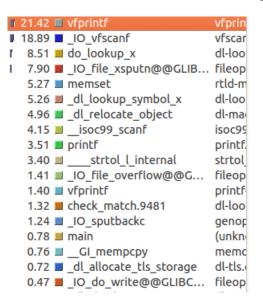


Figure 3: Annotated