



# Introduction to Computational Genomics

COMPSCI 260

## Graph search algorithms

Given a graph with vertices and edges, we often wish to determine a path from some start vertex to some end vertex within that graph. There are numerous ways to explore the vertices in a graph, but two popular methods are depth first search (DFS) and breadth first search (BFS). Upon closer examination, one can see that these two algorithms only differ in the data structure used to determine which vertex to visit next.

The pseudo-code for a general graph searching algorithm is given below:

```
GRAPH_SEARCH (Graph G, Vertex s)
  Set S = {s}; # set of explored vertices
  while (there is an unused edge (u,v) connected to any node u in S)
    follow edge (u,v) from u to v
    set edge (u,v) as used
    add v to S
  end while
end
```

In what follows, note that in a directed graph, the neighbors of a vertex are only those vertices where an edge points from the current vertex to them.

### Breadth first search (BFS)

Breadth first search uses a queue as the data structure for determining which vertex to visit next. Every time a new vertex is visited, all of its neighbors are added to the end of the queue. To decide which vertex to visit next, a vertex from the front of the queue is removed and the process is repeated. Thus, the vertices are visited in a first in first out (FIFO) manner. The result is that all of the vertices that are a distance  $i$  away from the start vertex are visited before the vertices that are a distance  $i+1$  away from the start vertex.

The pseudo-code for BFS is given below:

```
BFS (Graph G, Vertex s)
  Set S = {s}; # set of explored vertices
  Queue Q = all neighbors of s;
  while (Q is not empty)
    dequeue vertex v from the front of Q # shift in Perl
    if(v is not in S)
      add v to S
      enqueue neighbors of v onto the end of Q # push in Perl
    end if
  end while
end BFS
```

### Depth first search (DFS)

Depth first search uses a stack as the data structure for determining which vertex to visit next. Every time a new vertex is visited, all of its neighbors are added to the top of the stack. To decide which vertex to visit next, a vertex from the top of the stack is removed and the process is repeated. Thus, the vertices are visited in a last in first out (LIFO) manner. The result is that vertices are visited down one path from the start vertex until the path can be extended no longer, then another path is visited from the start vertex until it can be extended no longer, etc.

The pseudo-code for DFS is given below:

```
DFS (Graph G, Vertex s)
  Set S = {s}; # set of explored vertices
  Stack T = all neighbors of s;
  while (T is not empty)
    pop vertex v from the end of T # pop in Perl
    if (v is not in S)
      add v to S
      push neighbors of v onto the end of T # push in Perl
    end if
  end while
end DFS
```

### Sources:

Josh Robinson's notes for CPS160 (Spring 2005)

Chapter 22 of the second edition of **Introduction to Algorithms** (Second edition) by Cormen, Leiserson, Rivest and Stein (CLRS).