Breadth-First Search

Input: An unweighted graph and a start vertex u

Idea:

- Maintain a set R of vertices that have been reached but not searched and
- a set S of vertices that have been searched.
- The set R is maintained as a First-In First-Out list (queue)

Initialization: $R = u, S = \varnothing, d(u, u) = 0$

Iteration: As long as $R \neq \varnothing$, we search from the first vertex v of R. The neighbors of v not in $S \cup R$ are added to the back of R and then v is removed from the front of R and placed in S.

Example

Let G be the adjacency graph of the following Go shape:

```
  1  2  3  4  5  6
1    .  o  o  o  .
2    .  .  .  .  .  <--- black component
3    .
```

v	R	S
(2,1)	[(2,1)]	{}
(2,1)	[(2,2)]	{(2,1)}
(2,2)	[(2,3),(3,2)]	{(2,1),(2,2)}
(2,3)	[(3,2)]	{(2,1),(2,2),(2,3)}
(3,2)	[(4,2)]	{(2,1),(2,2),(2,3),(3,2)}
(4,2)	[(5,2)]	{(2,1),(2,2),(2,3),(3,2),(4,2)}
(5,2)	[(6,2)]	{(2,1),(2,2),(2,3),(3,2),(4,2),(5,2)}
(6,2)	[(6,1)]	{(2,1),(2,2),(2,3),(3,2),(4,2),(5,2),(6,2)}
(6,1)	[]	{(2,1),(2,2),(2,3),(3,2),(4,2),(5,2),(6,2),(6,1)}

References

- West. D. Introduction to Graph Theory (second edition) 2.3.8