

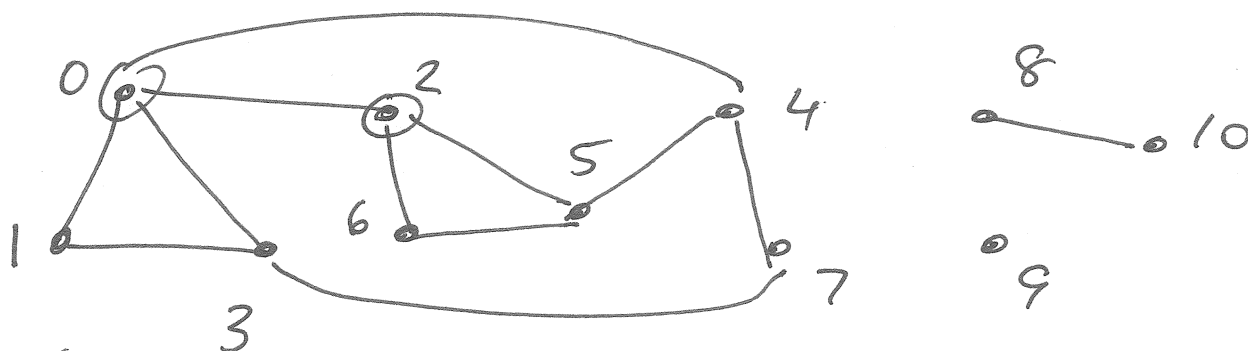
Intro to C++ Comp Prog 6/16/26

Breadth First Search

Input: unweighted, undirected graph
and vertex to start from

Goal: ① Mark the connected component
the vertex belongs to

② Calculate the distance (in # edges)
of the shortest path from the start
vertex to all the others.



BFS(6, 2)

	1	0	1	2	3	4	5	6	7	8	9	10
dist	∞	∞	0	∞	2	∞	∞	∞	∞	∞	∞	∞

queue: 2, 8, 8, 6, 1, 3, 4, 7

enqueue
→ adds to back
dequeue
→ removes from
front

push
pop
front
size

1. Set up dist array, add source queue

2. while queue isn't empty

$v = \text{dequeue next elem}$ // $v \rightarrow \text{next edge}$

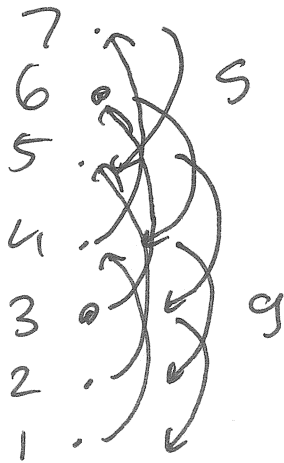
for each neighbor next of v

if $\text{dist}[\text{next}] == \infty$

$\text{dist}[\text{next}] = \text{dist}[v] + 1$

enqueue(next)

elevator trouble



$$U = 3$$
$$d = 2$$

1	2	3
4	5	
7	8	6



1	2	3
4		5
7	8	6



1	2	3
	4	5
7	8	6

Not all BFS problems have an explicit graph in the description. Often times, the graph structure is "implicit" or "not obvious". In code, we never build the full graph.