

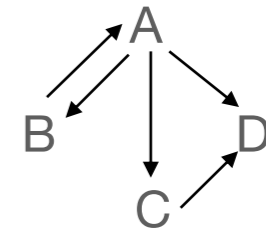
[320] Search Order and Queue Structures

Meenakshi Syamkumar

Tracing DFS

```
def find(self, dst):  
    self.graph.visited.add(self)  
  
    if self == dst:  
        return (self.name,)  
  
    for child in self.children:  
        if not child in self.graph.visited:  
            childpath = child.find(dst)  
            if childpath:  
                return (self.name,) + childpath  
  
    return None
```

A.find(D)



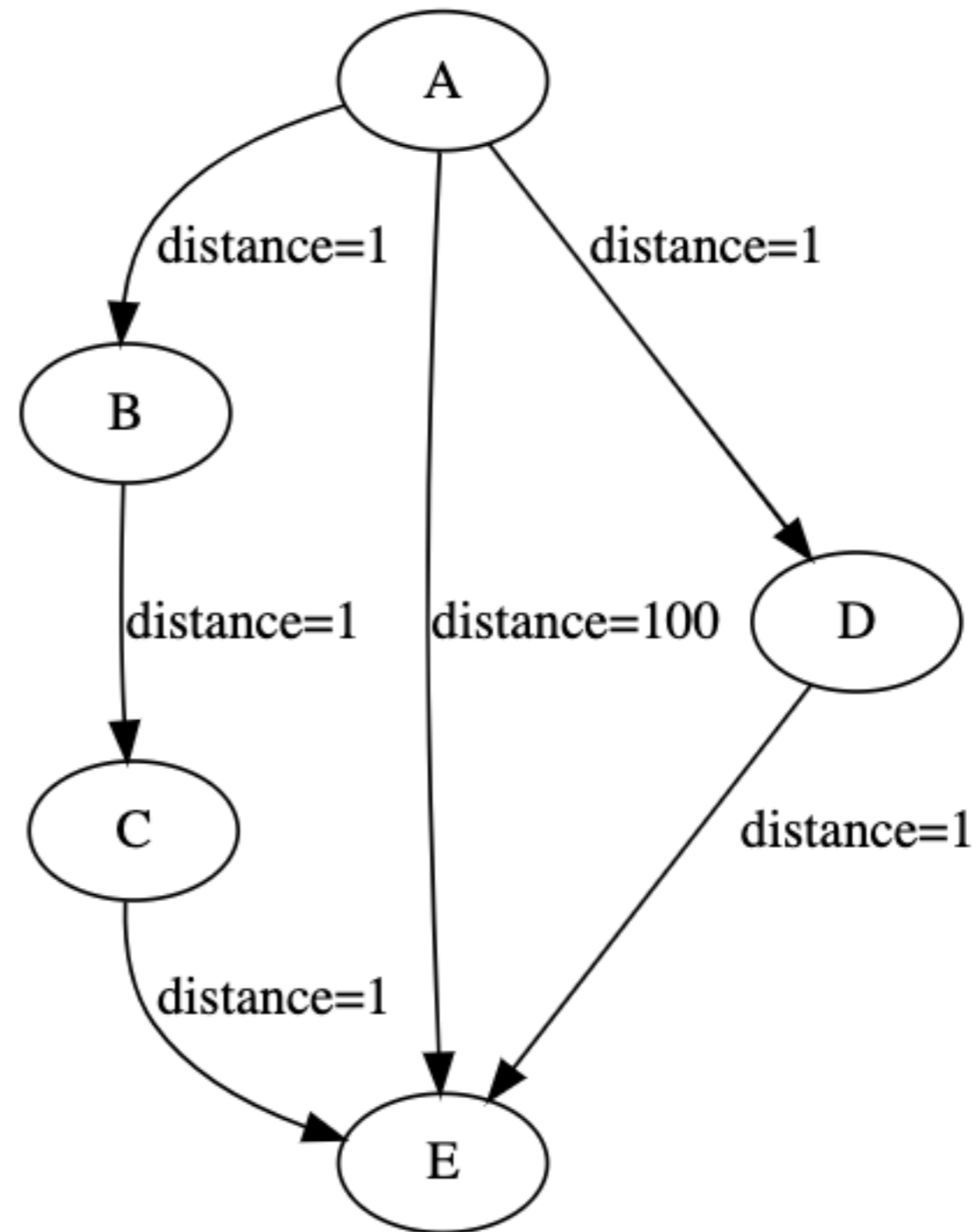
Queueing Structures

Shortest Weighted Path

What path will DFS choose?

What path will BFS choose?

What path would you choose?



Work queues ("to do" list)

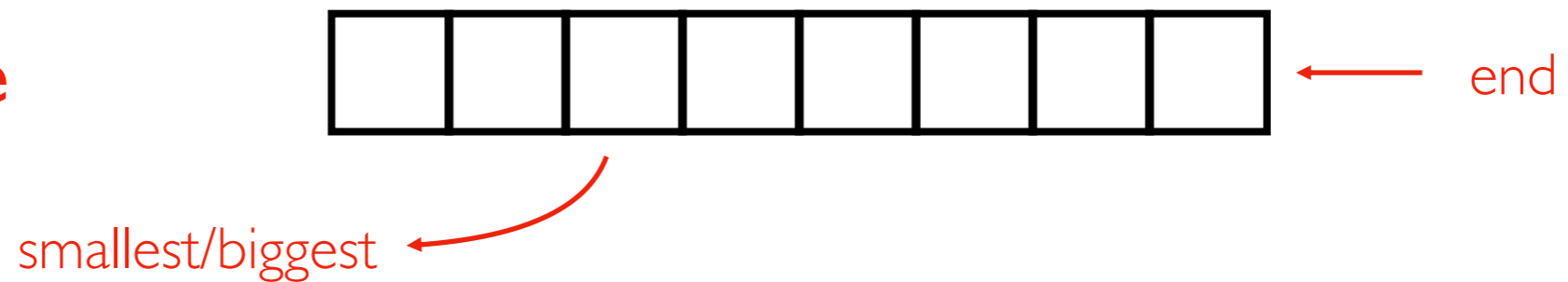
Stack
(FILO)



Queue
(FIFO)



Priority Queue



Work queues ("to do" list)

Stack
(FILO)



```
L.append(x)
```

```
x = L.pop(-1)
```

Queue
(FIFO)

front ←



```
x = L.pop(0)
```

```
L.append(x)
```

Priority Queue

smallest/biggest ←



```
L.sort()  
x = L.pop(-1)
```

```
L.append(x)
```

what operations are slow?

Work queues ("to do" list)

Stack
(FILO)



```
L.append(x)
```

```
x = L.pop(-1)
```

Queue
(FIFO)

front ←



!

```
x = L.pop(0)
```

```
L.append(x)
```

Priority Queue

smallest/biggest ←



!

```
L.sort()  
x = L.pop(-1)
```

```
L.append(x)
```

what operations are slow?