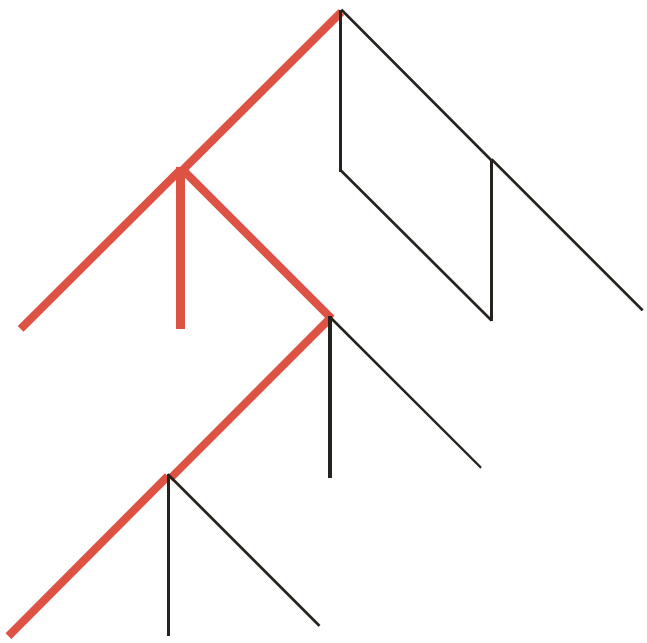
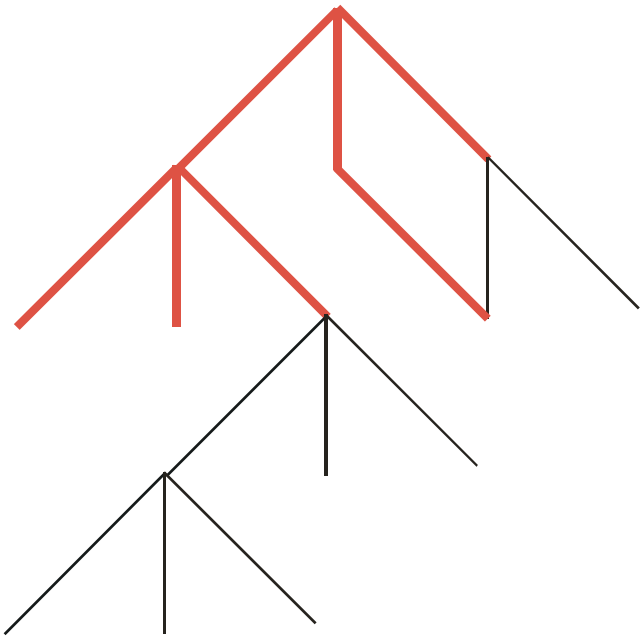
DFS vs BFS





```
DFS(G,v)      { using a stack S }
push(S,v)
WHILE NOT empty(S) DO
•     current ← pop(S)
•     mark(current) ← true
•     FOR u ∈ neighbor(G,current)
•     •     IF NOT mark(u) THEN push(S,u)
```

DFS vs BFS

```
BFS(G,v)      { using a queue Q }
enqueue(Q,v)
WHILE NOT empty(Q) DO
•     current ← dequeue(Q)
•     mark(current) ← true
•     FOR u ∈ neighbor(G,current)
•     •     IF NOT mark(u) THEN enqueue(Q
```

DFS vs BFS

DFS(G,v) { using a stack S }

mark(v) ← true; push(S,v)

WHILE NOT empty(S) **DO**

- current ← pop(S)
- **FOR** u ∈ neighbor(G,current)\{current}
- • **IF NOT** mark(u) **THEN**
- • • mark(u) ← true;
- • • mark((current,u)) ← discovery;
- • • push(S,u)
- • **ELSE** mark((current,u)) ← back

DFS vs BFS

```
BFS(G,v)      { using a queue Q }
mark(v) ← true; enqueue(Q,v)
WHILE NOT empty(Q) DO
•     current ← dequeue(Q)
•     FOR u ∈ neighbor(G,current)\{current}
•     •     IF NOT mark(u) THEN
•     •     •     mark(u) ← true;
•     •     •     mark((current,u)) ← discovery;
•     •     •     enqueue(Q,u)
•     •     ELSE mark((current,u)) ← back;
```