Graph Explore (Traversal)

Depth-first search (DFS)

```
procedure DFS(G,v):
    label v as discovered / visisted
    for each neighboring vertex w of v do
        if vertex w is not yet labeled as discovered / visited then
            recursively call DFS(G,w)
```

Property (fact):

- For any vertex $w \in G$,
  if there exists a path between v and w, then w will be labelled visited after a call to DFS(G,v).

  (why?)

Breadth-first search (BFS)

```
procedure BFS(G,v):
    let S be an empty queue (doubly-linked list)

    label v as discovered
    add v to the tail of S

    while S is not empty
        Extract the first element of S, let it be u.

        for each neighboring vertex w of u do
            if w is not labeled as discovered then
                label w as discovered
                mark the parent of w to be u
                add w to the tail of S
```

Property (fact):

- For any vertex $w \in G$,
  if there exists a path between v and w, then a call to BFS(G,v) will visit w using the shortest path between them.
  (亦即，呼叫 BFS(G,v) 時，從 v 拓展至 w 的過程，必定會對應到它們之間的最短路徑)

  (why?)