Problem:

給定一 $G=(V,E)$ 為一 digraph 和一 $s \in V$, s 為 source vertex

欲以 breadth-first 方式走訪 G 上 s 可走訪之點

建構以 s 為 root 之 BFS tree

idea: 用一 Queue 方式儲存所有已走訪但仍有相鄰點尚未走訪的點

定義: $\forall u, v \in V$

①. $u.color = \begin{cases} \text{white} \\ \text{gray} \\ \text{black} \end{cases}$ 作為狀態

②. $u.d$ = s 至 u 之 BFS distance

③. $\delta(s,v)$ = s 至 v 之 SP

Time Complexity: n個 vertex 進到 queue 一次 $\Rightarrow O(|V|)$

for loop $v \in adj[u] \Rightarrow O(|V|)$

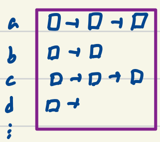
$\Rightarrow O(|V|^2)$

但只有 K_n 的 for loop 會為 $O(|V|)$

(1). adjacency matrix:

$\rightarrow O(|V|)$ $\Rightarrow O(|V|^2)$

(2). adjacency list:



$\Rightarrow O(|V|+|E|)$

$O(2|E|)$

Note: ①. 上述版本之 BFS 只能走訪 s reachable 之 vertex

↳ 可用作 check $u \in V$ 下 u 之 reachable 之 vertex

BFS algorithm:

```
for each v ∈ G.V - {s}
    v.π = NiL
    v.d = ∞
    v.color = w
s.color = g
s.π = NiL
s.d = 0
Let Q be a queue
Enqueue(Q, s)
while Q ≠ ∅
    u = dequeue(Q)
    for each v ∈ G.adj[u]
        if v.color == w
            v.color = g
            v.d = u.d + 1
            v.π = u
            enqueue(Q, v)
    u.color = black
```

②. print 出 BFS(G, s) 後 s 至任一 $v \in V$ 之 path

```
print_path(G, s, v)
if s == v
    print(s)
else print_path(G, s, v.π)
    print(v)
```

應用：① 判斷圖是否為 connected：

Algo：1. 任選一 vertex $u \in G.V$

2. BFS(G, u)

3. for each $v \in G.V$

if v.color != black

return False

$O(|V|+|E|)$

② 求 unweighted graph 上之 SP

$\because$ BFS 跑完後, $\forall u \in G.V$, $u.d = \delta(s, u)$

③ diameter： $\max_{u,v \in V} \delta(u,v) \Rightarrow$ 求 all-pairs SP 後取 max $\Rightarrow O(|V|^2+|V||E|)$

④ Tree 上找 diameter

⑤ 找 transitive closure

定義為：$G^* = (V, E^*)$

其中：$E^* = \{(V_i, V_j) \mid V_i, V_j \in V$ 且 G 上存在 V_i 至 V_j 之 path$\}$

正確性：BFS(G, s) 後 s 之 non-reachable 之 vertex 為 $v.d = \infty$

$\therefore$ for each $v \in G.V$

BFS(G, v)

for each $u \in G.V$

if $u.d < \infty$

T[v, u] = 1

else

T[v, u] = 0