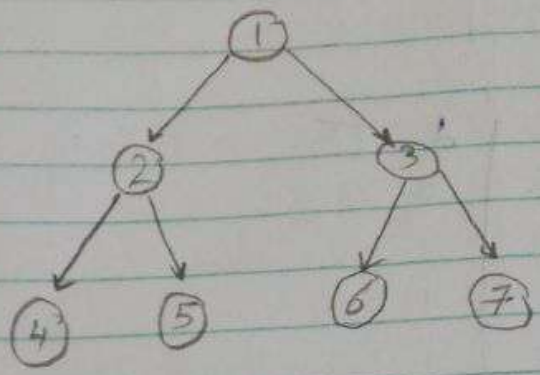
الاسم: هاشم عادل سعيد
ID: 20210269

Graph - HW#5

Q(1):



- list

1 → 2 → 3
2 → 4 → 5
3 → 6 → 7
4
5
6
7

- Matrix

	1	2	3	4	5	6	7
1	0	1	1	0	0	0	0
2	0	0	0	1	1	0	0
3	0	0	0	0	0	1	1
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0

(2) Date 20210269 No

Q(2):

	d	π
1	∞	null
2	3	4
3	0	null
4	2	5
5	1	3
6	1	3

queue
3 ↓
5
6
4
2
1

Q(3):

	d	π
u	0	null
t	1	u
x	1	u
y	1	u
s	3	w
w	2	t
r	4	~~s~~
v	5	r

queue
u ↓
t
x
y
w
s
r
v

Q(4):

	d	π
S	0	null
A	1	S
D	1	S
E	2	D
B	2	A
C	3	B
F	3	E
G	4	F

(8) 2021-269

Q(5):

```
DFS(matrix, start)
{
    n ← len(matrix)
    stack.start
    visited[start] = true
    while stack
    curr = stack.Pop()

for i ← n-1 to 0 by -1 do
if matrix[curr][i] == 1 and visited[i] is false
    stack.append(i)
    visited[i] = true
end if
end for
end while
}
```

(4) 2021026

Q(7):

	Π	dt	Ft
c	Null	1	20
b	c	2	11
G	b	3	10
F	G	4	9
a	F	5	8
e	a	6	7
i	c	12	17
h	i	13	14
j	i	15	16
d	c	18	19

Q(9):

- The running time of BFS with an adjacency matrix is:

- Worst Case: ~~$O(V+E)$~~ $O(V^2)$
- ~~Most Case~~: ~~$O(V+E)$~~ $O(V^2)$
  Average Cases

Best Case: $O(V+E)$

(5) 20210269

```
BFS (Matrix, start)
{
    n ← length (matrix)
    Visited [n] = False
    queue [rear] = start
    visited [start] = true
    while front <= rear
        curr = queue [front++]
        For i ← 0 to n by i++
        if matrix [curr] [i] = 1 and ! visited [i]
            queue [++rear] = i
            visited [i] = true
        end if
        end for
        end while
}
```