



Graph

สื่อการสอนสำหรับค่ายคอมพิวเตอร์โอลิมปิก สอน. ค่าย 2/2562

โดย นางสาว นันก์ณกัส บันลือสมบัติกุล, สถาบันวิทย์สีรีเมร์ (VISTEC)

ห้ามนำสื่อการสอนนี้ดู ไปใช้โดยไม่ได้รับอนุญาตจากผู้จัดทำ

<http://www.free-powerpoint-templates-design.com>



สวัสดี

นันก์กุ๊ส บันลือสมบัติกุล (แบบ)

จบปริญญาตรี สาขาวิทยาการคอมพิวเตอร์
มหาวิทยาลัยธรรมศาสตร์

ปัจจุบัน นักศึกษาปริญญาเอก
สาขา Information Science and Technology
สถาบันวิทย์สีรีเมค (VISTEC)



Agenda

01

ทำความรู้จักกับกราฟ

กราฟคืออะไร ตัวอย่าง ประโยชน์

02

นิยามกราฟ

การพูดถึงกราฟด้วยตัวโปรแกรมคณิตศาสตร์

03

การสร้างกราฟ

วิธีการสร้างกราฟ, Coding, Exercise

04

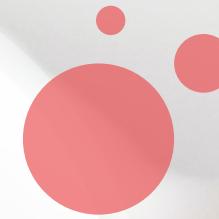
การท่องกราฟ

Algorithm พื้นฐานสำหรับกราฟ (BFS, DFS)

การท่องกราฟ

Algorithm พื้นฐานสำหรับกราฟ (BFS, DFS)





เช็คชื่อกันเดอะ

<https://forms.gle/NVVB3UWFnuxsxnXA>

การท่องกราฟ

Breadth First Search or BFS, Depth First Search or DFS



- กำหนด: Graph $G = (V, E)$, จุดเริ่มต้น (**Source**)

การท่องกราฟ คือ การหา path ที่เริ่มจากจุดเริ่มต้น ไปจนครบทุก node บนกราฟ
มี 2 วิธีหลักๆ คือ

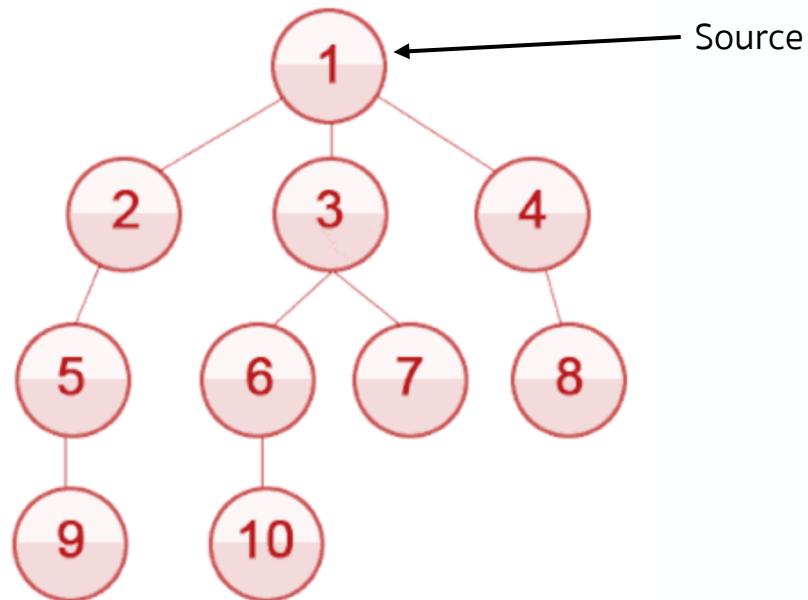
- Breadth First Search (BFS) การเดินในแนวกว้าง
- Depth First Search (DFS) การเดินในแนวลึก

การท่องกราฟ

Breadth First Search or BFS



- Breadth First Search (BFS) การเดินในแนวกว้าง



การท่องกราฟ

Breadth First Search or BFS



- **Breadth First Search (BFS) การเดินในแนวกว้าง**

- ใช้ list มาเก็บ queue ของ node ที่เราต้องไป และ Boolean array มาช่วยเก็บว่า เราเดินผ่าน node มาแล้วหรือยัง

BFS: (For graph G consisting V nodes)

initialize all in **visited[V]** = false, empty **queue**;

mark **visited[source]** = true;

push **source** to **queue**;

while (**queue** is not empty) {

s = **queue.pop_front()**;

show **s**;

for each **node** in **adjacency(s)** {

if **visited[node]** == false {

visited[node] = true;

queue.push_back(node);

} //end if

} //end for

} //end while

การท่องกราฟ

Breadth First Search or BFS

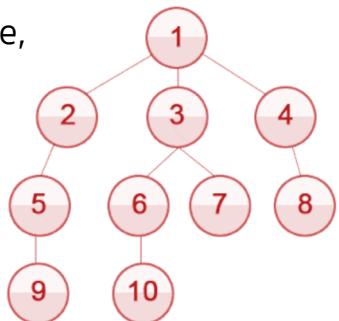


- **Breadth First Search (BFS) การเดินในแนวกว้าง**

initialize all in **visited[V]** = false,
empty **queue**;

mark **visited[source]** = true;
push **source** to **queue**;
while (**queue** is not empty) {

```
s = queue.pop_front();
show s;
for each node in adjacency(s) {
    if visited[node] == false {
        visited[node] = true;
        queue.push_back(node);
    } //end if
} //end for
} //end while
```



1	2	3	4	5	6	7	8	9	10
visited	F	F	F	F	F	F	F	F	F

queue → { }

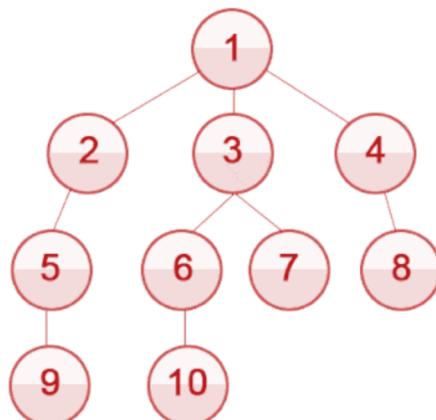
output

การท่องกราฟ

Breadth First Search or BFS



- แบบฝึกหัด ให้บันทึกผลลัพธ์ที่ได้จาก BFS เมื่อเริ่มจาก node 3



	1	2	3	4	5	6	7	8	9	10
visited	F	F	F	F	F	F	F	F	F	F

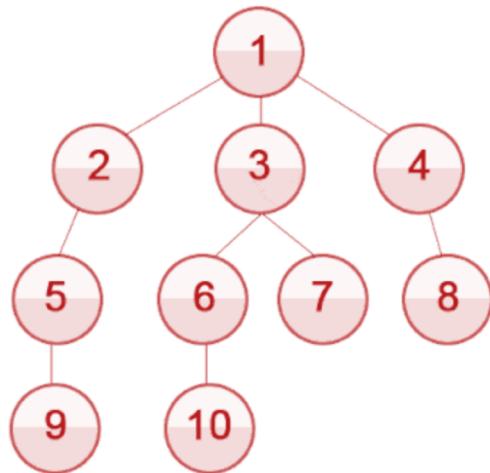
queue → { }

output

การท่องกราฟ

Breadth First Search or BFS

- Breadth First Search (BFS) การเดินในแนวกว้าง



Let's code !!

[03-BFS.cpp]

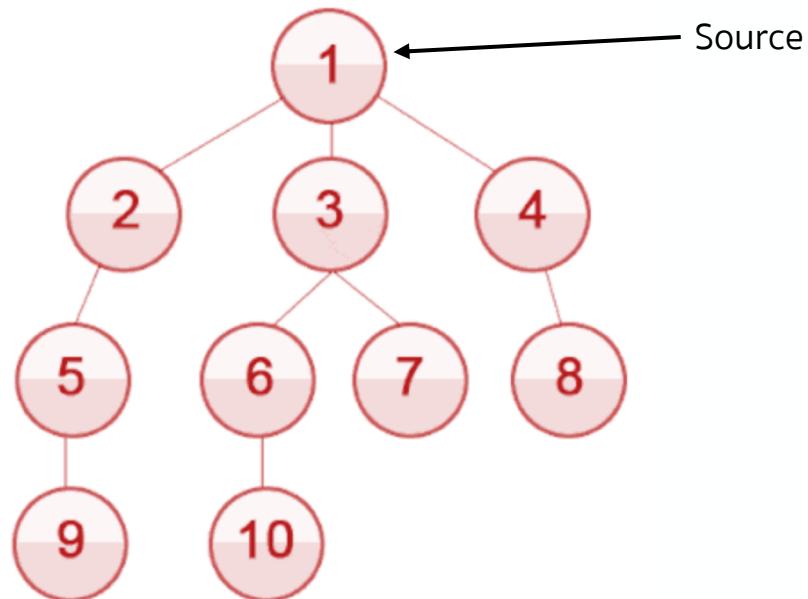


การท่องกราฟ

Depth First Search or DFS



- Depth First Search (DFS) การเดินในแนวลึก



การท่องกราฟ

Depth First Search or DFS



- **Depth First Search (DFS) การเดินในแนวลึก**

- ใช้ Boolean array มาช่วยเก็บว่า เราเดินผ่าน node มาแล้วหรือยังเช่นกัน แต่ไม่มี queue ใช้การ recursive แทน

(For graph G consisting V nodes)

DFS:

initialize all in **visited[V]** = false,

//call recursive function

DFS_util(source, visited);

DFS_util(v, visited):

show v;

mark **visited[v] = true;**

For each **node** in adjacency(**v**):

if **visited[node] == false:**

DFS(node, visited);

การท่องกราฟ

Depth First Search or DFS



- Depth First Search (DFS) การเดินในแนวลึก

- ใช้ Boolean array มาช่วยเก็บว่า เราเดินผ่าน node มาแล้วหรือยังเช่นกัน แต่ไม่มี queue ใช้การ recursive แทน

(For graph G consisting V nodes)

DFS:

initialize all in **visited[V]** = false,

//call recursive function

DFS_util(source, visited);

DFS_util(v, visited):

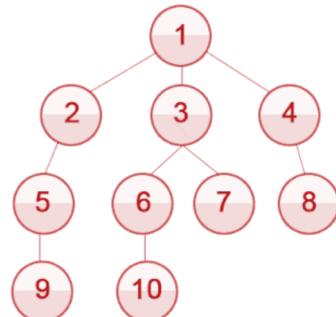
show v;

mark **visited[v] = true**;

For each **node** in adjacency(**v**):

if visited[**node**] == false:

DFS(node, visited);



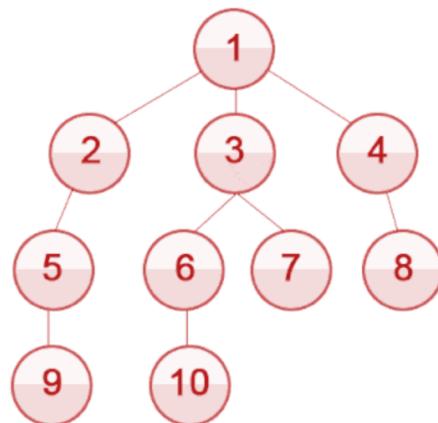
output

การท่องกราฟ

Depth First Search or DFS



- แบบฝึกหัด ให้บันทึกผลลัพธ์ที่ได้จาก DFS เมื่อเริ่มจาก node 3



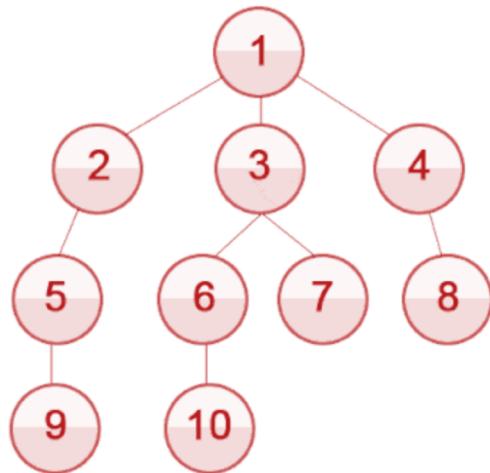
	1	2	3	4	5	6	7	8	9	10
visited	F	F	F	F	F	F	F	F	F	F

output

การท่องกราฟ

Depth First Search or DFS

- Depth First Search (DFS) การเดินในแนวลึก



Let's code !!

[04-DFS.cpp]



ກາຣທົ່ວກຮາວ

Depth First Search or DFS



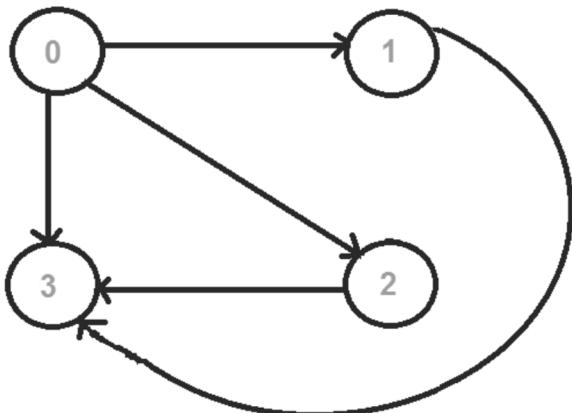
Quiz ☺

<https://forms.gle/pFbbsG4bCJPGcu7E6>

การท่องกราฟ

Depth First Search or DFS

- แบบฝึกหัด: จงหาว่า Node ใดเดินถึงกันได้บ้าง โดยใช้ DFS



- ข้อมูลนำเข้า (Input)

- จำนวน Vertex (N)
- จำนวน Edge (E)
- ต้นทาง ปลายทาง ของ เส้นทางที่ 1
- ต้นทาง ปลายทาง ของ เส้นทางที่ 2
- ...
- ต้นทาง ปลายทาง ของ เส้นทางที่ E

- ตัวอย่าง input

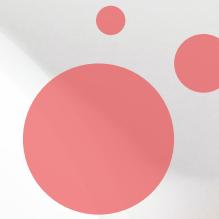
4
5
0 1
0 2
0 3
1 3
2 3

- ผลลัพธ์ (Output)

- Matrix ขนาด $N \times N$ ที่ 1 แทนการเดินถึงกับได้ และ 0 แทนเดินถึงกันไม่ได้

- ตัวอย่าง output

[[1 1 1 1]]
[0 1 0 1]
[0 0 1 1]
[0 0 0 1]]



Thank you

ສູ້າ ບະຄະທຸກຄນ