

---

---

---

---

---

---

---



## Minimum Spanning tree

- Kruskal  $\rightarrow$  DSU + เรียงตาม + heap  
เพื่อเลือกที่ไม่เกิด cycle

- Prim  $\rightarrow$  เลือกโหนดใน set แล้วหา + heap (เหมือน Dijkstra) + check  
(รวม node ที่ติดกับ set) ใน set ใหม่

## Shortest path (จุดไปจุด)

- BFS = weight มีเท่ากัน  
= queue รับมา push ไปยัง + mark

- Dijkstra = weight ไม่เท่า  
= heap + mark แล้ว push

- Bellman ford

ไม่มีสิ่งกีดขวาง  
(imprison) พม่า  
ทำท่าว่าสู้กัน  
กระแทกกันแล้ว  
กันเป็นท่อนั้น  
(เรื่อกว่า)

## Shortest path จากจุดไปทุกจุด

Floyd - Warshall Algorithm [all pairs shortest path]

1.  $dis[i][j]$   $\hookrightarrow O(n^3)$

2. set  $dis$  ทุกอันเป็น 0 (ถ้าเป็น  $dis[i][i][i]$  set = 0)

3. set ขอบเส้นทาง ( $dis[u][v] = dis[v][u] = 1$ )

4. จน 3 จบทำกัน  
 $dis[i][j] = \min(dis[i][j], dis[i][k] + dis[k][j])$

## Topological Sort

- Recursive DFS  
(จากมา sort จากหน้าไปหลัง)

- Kahn's algorithm  
push node ที่ไม่มี degree in heap

## State Dijkstra

Ex: locked room, Teap's nitrous

push node ที่ติดกับ lock  
หรือ nitrous

heap เพื่อเก็บค่าที่น้อย

Ex: locked room  $\rightarrow$  switch 1,0

Teap's nitrous  $\rightarrow$  นิตรัส 1,0

push ไปยัง 2 นิตรัส

mark ที่ push 2 นิตรัส

## LIS (Longest Increasing Substring)

$O(n \log n)$

int dp[50000];

int main()

int n, idx, cnt=0, num;

scanf("%d", &n);

for (i=0; i<n; i++)

scanf("%d", &num);

num[idx] = upper\_bound(dp, dp+cnt, num) - dp;

if (idx > 1 && dp[idx-1] == num) continue;

if (idx == cnt) cnt++;

dp[idx] = num;

num[idx] = lower\_bound(dp, dp+cnt, num) - dp;

if (idx == cnt) cnt++;

dp[idx] = num;

};

printf("%d", cnt);

## Sort

1. count sort  $\rightarrow$  นับว่าเลขมีกี่ตัว  $AC[ ] =$  จำนวน  
(เก็บเลขแล้วเก็บไว้ที่  $AC$ )

2. merge sort  $\rightarrow$  แบ่งเป็นครึ่ง  $\rightarrow$  ทำทุกครึ่ง  $\rightarrow$  รวม

3. sort  $\rightarrow$  sort ( $a+1, a+n+1, greater<int>()$ )

## O

10, 15  $\rightarrow n!, 2^n$

50  $\rightarrow n^4$

log  $\rightarrow$  heap, bsearch

100-300  $\rightarrow n^3$

1000-5000  $\rightarrow n^2$

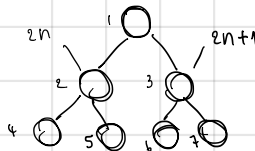
100000-500000  $\rightarrow n \log n, n$

$10^9 \rightarrow \log n$

$10^9 \rightarrow \text{not } n$

## Binary Search tree

ดูซ้ายบ้นขวากว่า หรือ ขอบ้นขวากว่า



## Bipartite graph

ถ้าเป็น bipartite (check)

ใช้ dfs หรือ bfs

ถ้า return 0 ก็คือ bipartite

int dfs (int u, int col) {

if (color[u] != 0 && color[u] == col) return 1

if (color[u] != 0 && color[u] != col) return 0

color[u] = col;

for (auto x : g[u]) {

if (dfs(x, 3-col) == 0)

return 0;

}

return 1;