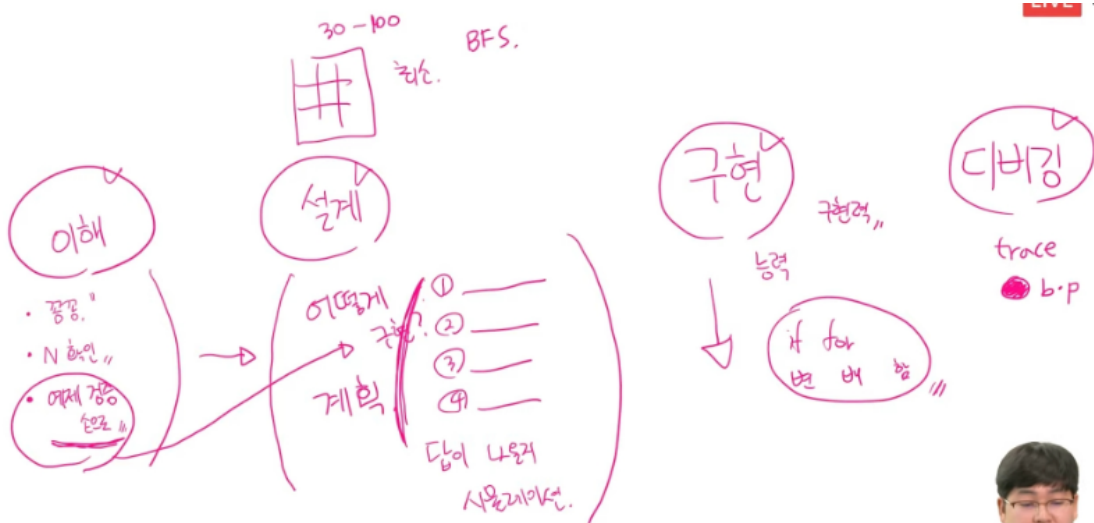
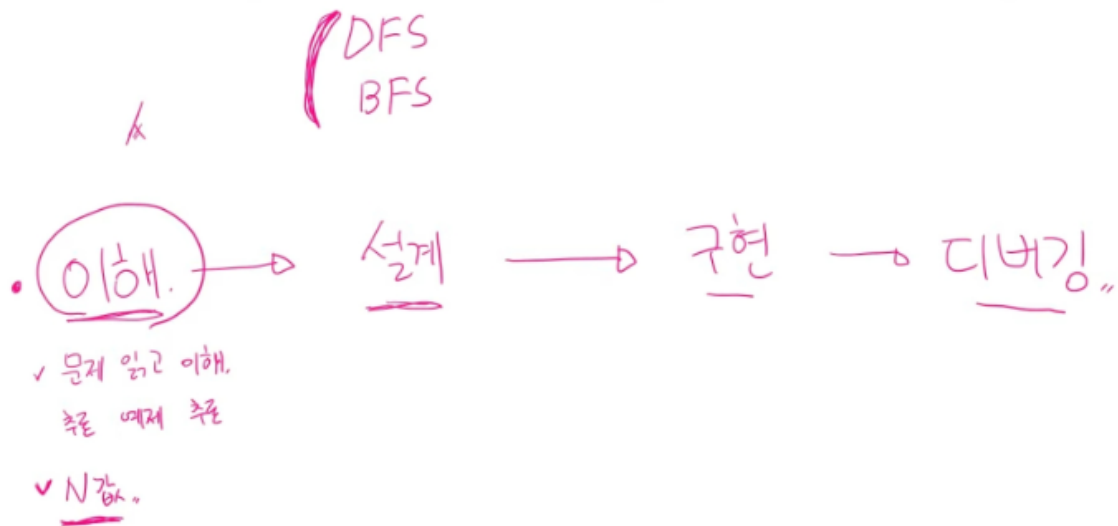


# A형 강의

문제 푸는 과정은 이해(N값 추론) 설계(tc로 단계별로 설계) 구현(코드 구현) 디버깅



DFS : n의 값이 1~30이면, BFS: n이 2차원이고 50~200이면 가능성이 높음

① 1 ~ 30 DFS  
 ② N 2차원, 50 ~ 200, BFS  
 , N 50 ~ 200, BFS  
 , 한 가지

## 구현테스트 3가지

### 1. 리스트를 만들어서 수를 for문으로 돌려 채움

```

1  # easy버전 구현력
2  # 리스트를 만들어서 수를 포문 돌려서 채움
3  # 55 44 33 22 11 0
4
5  list = []
6
7  for i in range(5, -1, -1):
8      list.append(i*11)
9
10 print(list)
11

```

```

# easy버전 구현력
# 리스트를 만들어서 수를 포문 돌려서 채움
# 55 44 33 22 11 0

list = []

for i in range(5, -1, -1):
    list.append(i*11)

print(list)

```

## 2. 중복제거후 소팅

```
1 # 중복을 제거 하고 알파벳 순서대로 출력
2
3 lst= ['A', 'B', 'T', 'A', 'A', 'A', 'B', 'A', 'A']
4
5 lst = list(set(lst))
6
7 lst.sort()
8 print(lst)
```

```
# 중복을 제거 하고 알파벳 순서대로 출력
```

```
lst= ['A', 'B', 'T', 'A', 'A', 'A', 'B', 'A', 'A']
```

```
lst = list(set(lst))
```

```
lst.sort()
```

```
print(lst)
```

## 3. 재귀호출 (10, 9, 8, 7, 6, 6, 7, 8, 9, 10이렇게 나오도록)

```
gg.py - C:/Users/multicampus/AppData/Local/Programs/Python/Python39/gg.py (3.9.7)
File Edit Format Run Options Window Help

def run(n):
    if n == 5:
        return

    print(n)
    run(n - 1)
    print(n)

run(10)
```

```
IDLE Shell 3.9.7
File Edit Shell Debug Opti
Python 3.9.7 (t
064)] on win32
Type "help", "c
>>>
== RESTART: C:/I
10
9
8
7
6
6
7
8
9
10
>>>
```

```
5 def num(n):
6     if n == 5:
7         return
8     print(n, end=' ')
9     num(n-1)
10    print(n, end=' ')
11
```

num()

sol1 (1) x

C:\Users\star3\AppData\Local\Programs\Py

10 9 8 7 6 6 7 8 9 10

```
# 순서 10, 9, 8, 7, 6, 6, 7, 8, 9, 10
```

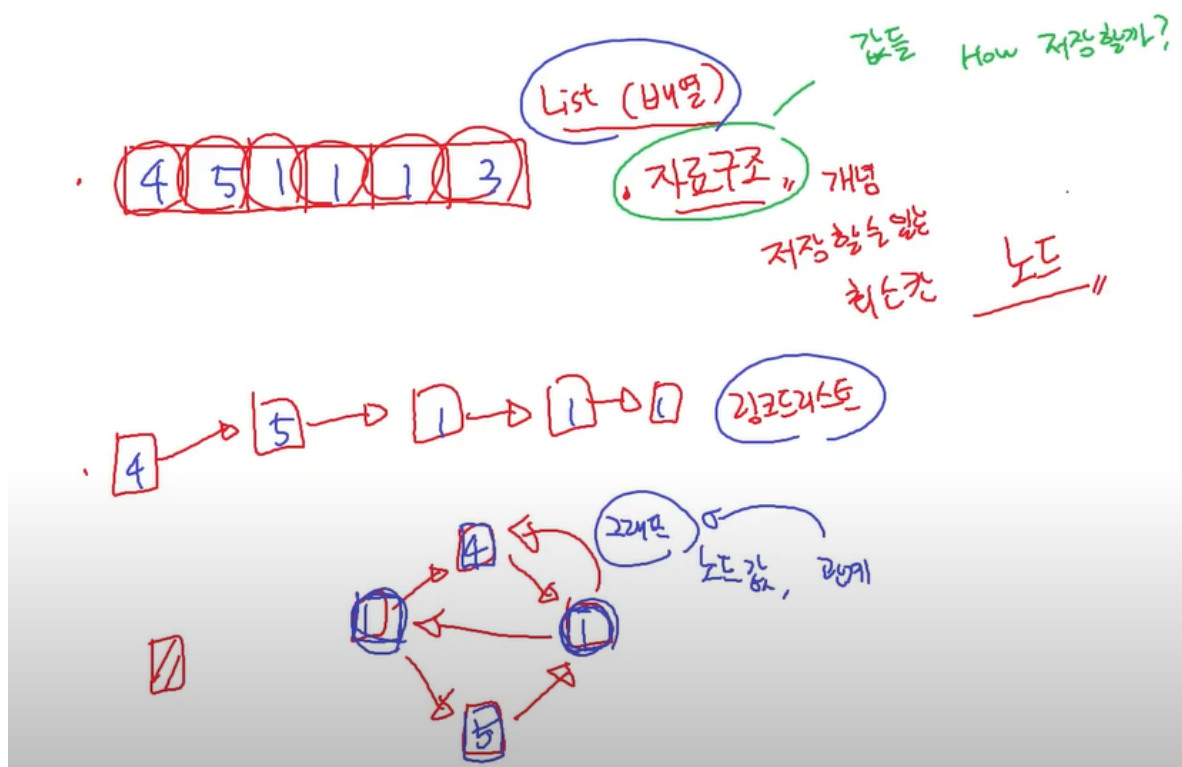
```
def num(n):  
    if n == 5:  
        return  
    print(n, end=' ')  
    num(n-1)  
    print(n, end=' ')
```

## BFS

리스트 자료구조에서 저장할 수 있는 최소칸을 노드라고 부름

값들을 어떻게 저장하고, 읽고 쓰고 관리할까?

리스트에 넣을까? 그래프에 넣을까? 연결리스트에 넣을까 등등을 고민함.



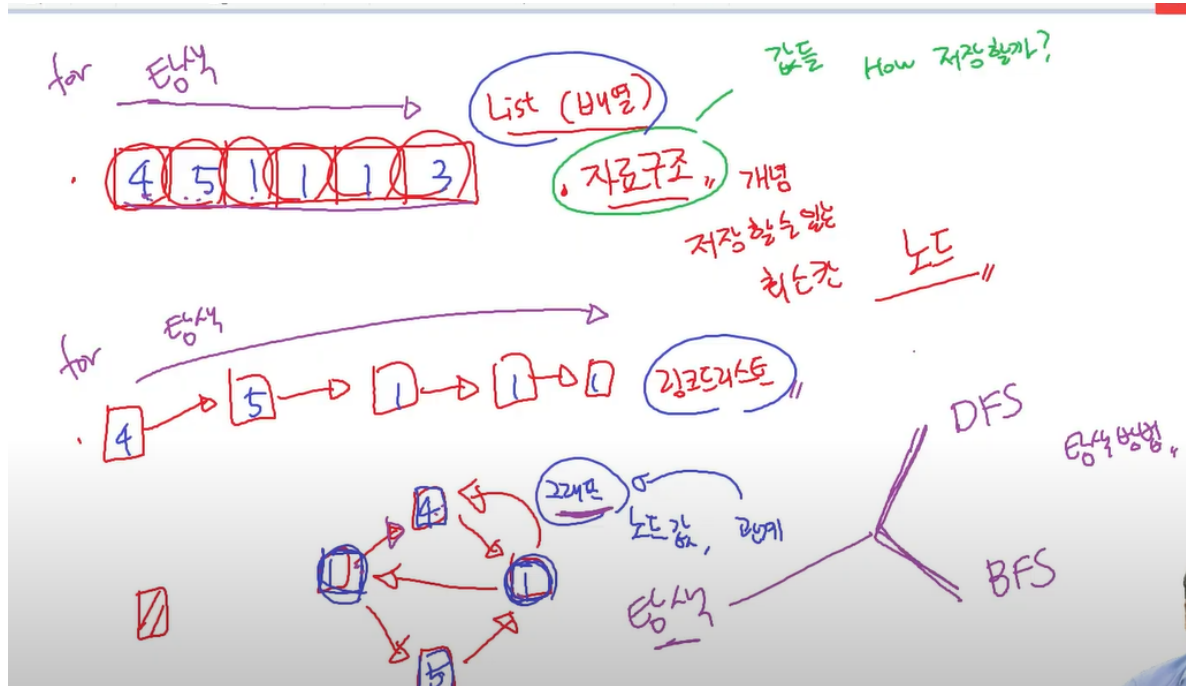
## 그래프를 탐색하는 방법 두가지가 BFS DFS

탐색 : 자료구조를 뒤지는 걸 탐색이라고 부름

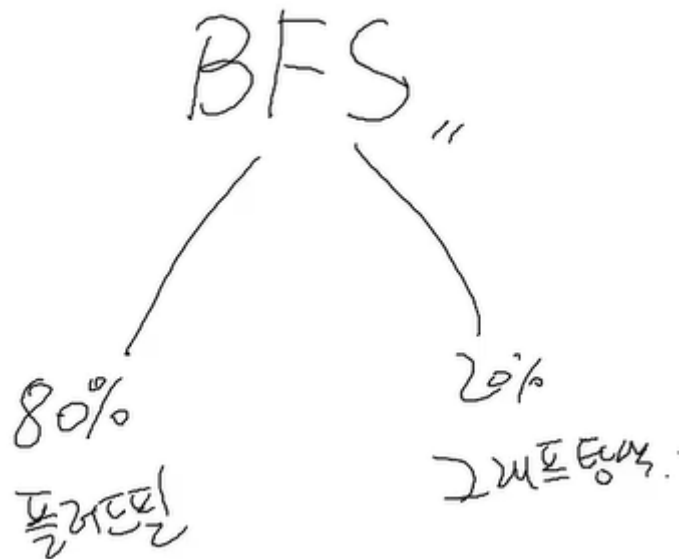
리스트, 연결리스트는 for로 탐색하지만

그래프는 for로 탐색하기가 힘들다.

그래서 그래프를 탐색하는 방법이 두가지가 있는데, 그게 바로 BFS DFS이다.



2차원배열에서 퍼지는 알고리즘 = 플러드필 알고리즘



이렇게 배열을 값으로 채워나가는 것을 플러드필 알고리즘이라고 부른다.

BFS의 일종임.



플러드 필 알고리즘

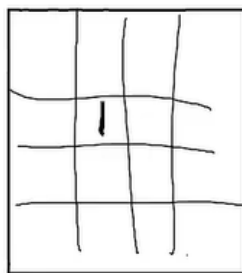
- 능숙도 테스트

## 문제

BFS..

기본 + ⑤

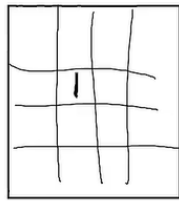
### 1. 1을 중심으로 퍼트리기(훈련하기)



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

BFS 기본.

Edge  
 - 시작점  
 - 끝점  
 :



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

BFS 기본.

- 방문  
 - 방문  
 :

for  $dydx \in (-1, 0), (1, 0), (0, 1), (0, -1)$

$ny = m + dy$

$nx = m + dx$

if ( $ny < 0$  ..... ) continue

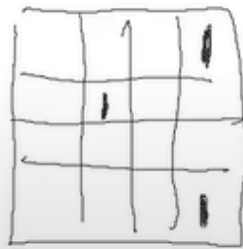
if ( $map[ny][nx] \neq 0$ ) continue.

map.

덱. push.C )

## 2. 1이 3개가 있고 2, 3을 넣기

② BFS 응용



3	2	2	4
2	1	2	2
3	2	2	3
3	2	1	2



for  $dydx$  in  $(-1,0), (1,0), (0,1), (0,-1)$

$ny = m + dy$   
 $nx = n + dx$

if  $(ny < 0 \dots)$  continue  
 if  $(map[ny][nx] \neq 0)$  continue.

map.  
 deque.push(( ))

① BFS 기본.

Edge  
 • bridge  
 • wall  
 :

[[1,1]]

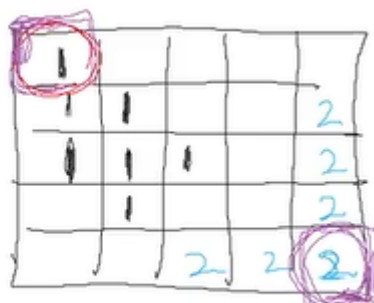
---

② BFS 응용

[(0,3), (1,1), (3,3)]

초기값이 [(0, 3), (1, 1), (3, 3)]

### 3. BFS 응용 : 천지창조 문제



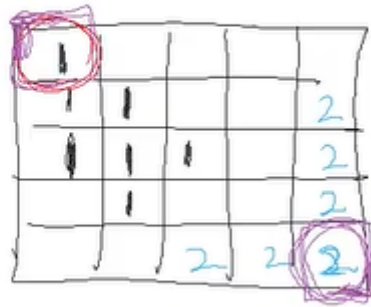
1의 크기

2의 크기

빨간 박스는 고정

등록할 때마다 카운팅하면 크기를 구할 수 있음

③ BFS 응용 성의 크기 "



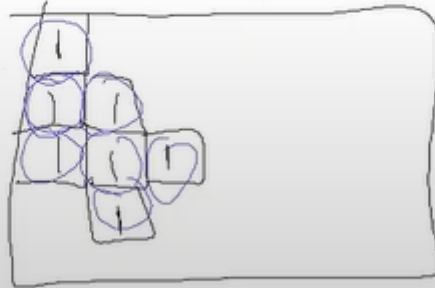
1의 크기

2의 크기

5분컷

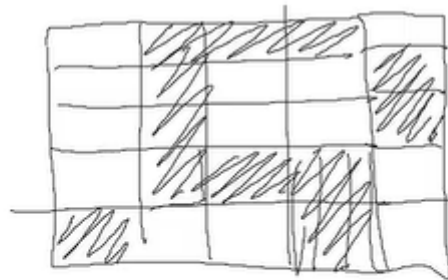
[[0,0]]

cnt++ 크기



#### 4. 섬 몇개? (5분컷)

④

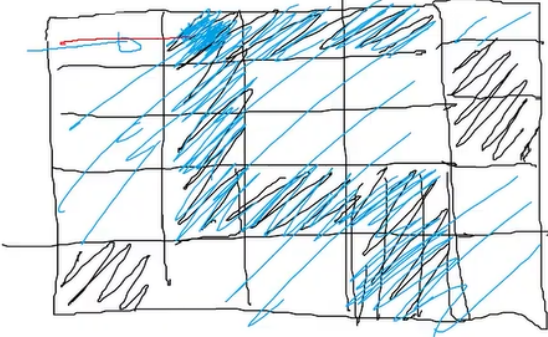


섬 몇개?

이중 for문을 돌려서 섬하나(한칸)를 찾음

bfs를 돌려서 섬을 제거함

④

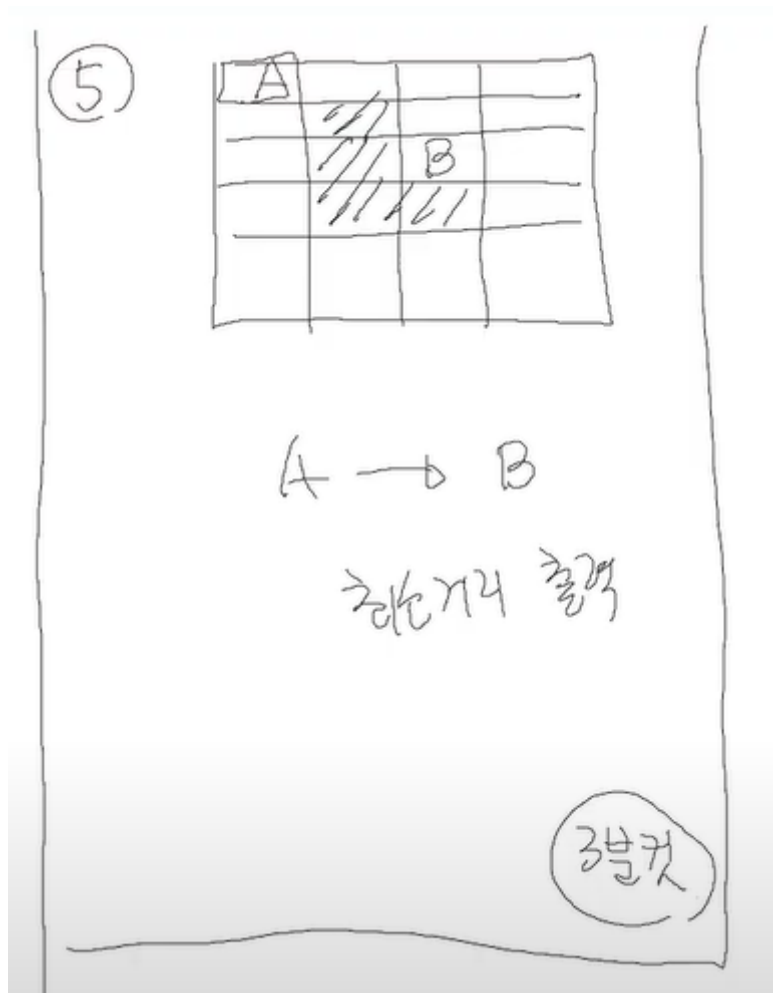


for →  
for →

성 하나 찾으면  
성 전체 제거  
cnt ++

성 몇개? 3개.

## 5. 미로찾기



0부터 채워나가서 발견되면 해당 값 출력

6.  $A \rightarrow B \rightarrow C$  최단거리로 가기 = 미로찾기 두번하기  $A \rightarrow B$  /  $B \rightarrow C$  (5분컷)

⑥

A	1	2	3
1	///	3	///
2	///	4	///
3	4	5	6B

A  $\xrightarrow{(6)}$  B  
 ↙  
 C  $\frac{1}{2}$ 만남.

⑥

A	1	2	3
1	///	3	///
2	///	4	///
3	4	5	6B

A  $\xrightarrow{(6)}$  B  
 ↙  
 C  $\frac{1}{2}$ 만남.

7.

## 미생물관찰 문제 D2

섬 개수 문제 (4번)

섬의 크기를 각각 수해서 max값 갱신

