배열 – 크기를 먼저 정해주고 변경 불가능 (값만 변경 가능) –속도가 좀 빠름

리스트 – 자유롭게 추가 삭제 가능 (크기 변경 가능)

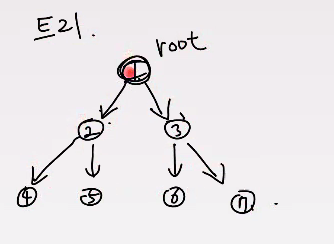
완전 탐색 – 모든 경우의 수를 보는 것 (코테에서 무조건 나옴)

구슬 탈출2 bfs+완전 탐색

Bfs, - 너비 우선 탐색(대부분의 길찾기 문제는 bfs)

dfs – 깊이 우선 탐색 (깊이가 깊은 문제는 시간 낭비일수도(문제에서는 10번만 하라했음) que로는 구현이 안됨

트리 –

시작 노드 - toot

무조건 1대일 이고 한방향으로 간다.

트리 구현. 이런걸 배열로 표현을 많이 한다.

A= [1,2,3,4,5,6,7]

A=

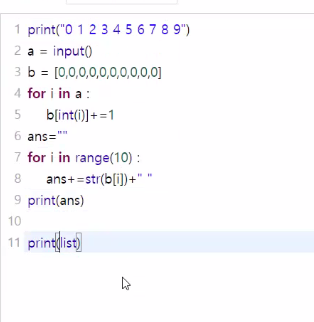
알고리즘을 사용하지 않아도 풀 수 있는게 삼성 문제

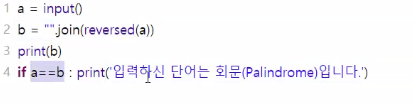
깊이 우선 탐

Split() : 띄어쓰기 기준으로

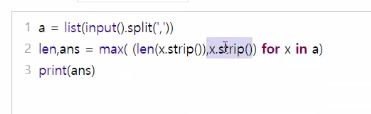
Strip() : 가져온 문자열을 한글자마다 나누어 주는거라고 생각

Visited = 이미 거쳐간 좌표 저장

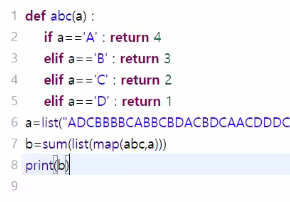




.join -> reversed객체는 str이 아니기 때문에 join 안쓰고 str써봐도 될 듯



strip 공백제거



보통 input()으로 받으면 시간초과

Import sys

Sys.stdin.readline()

**map함수 암기!**

**Dfs bfs**

**import sys**

**def dfs(i) :**

**if visited[i] : return**

**visited[i]=True**

**ans.append(i)**

**for j in graph[i] :**

**dfs(j)**

**n, m, v = map(int, sys.stdin.readline().split())**

**graph = [[] for \_ in range(n+1)]**

**for \_ in range(m) :**

**a,b = map(int, sys.stdin.readline().split())**

**graph[a].append(b)**

**graph[b].append(a)**

**visited = [0]\*(n+1)**

**for a in graph :**

**a.sort()**

**ans=[]**

**dfs(v)**

**print(\*ans)**

**ans=[]**

**visited = [False for \_ in range(n+1)]**

**que = [v]**

**while que :**

**next = que.pop()**

**if visited[next] : continue**

**visited[next]=True**

**ans.append(next)**

**for i in graph[next] :**

**que = [i]+que**

**print(' '.join(map(str,ans)))**

**2178 미로 탐색**

**import sys**

**n,m = map(int, sys.stdin.readline().split())**

**board=[]**

**for i in range(n) :**

**board.append(list(map(int, sys.stdin.readline().strip())))**

**q = [(0,0,1)]**

**visited = [[0 for \_ in range(m)] for \_ in range(n)]**

**dr = [0, 1, 0, -1] # 오른쪽, 아래, 왼쪽, 위쪽**

**dc = [1, 0, -1, 0]**

**while q :**

**now = q.pop()**

**for i in range(4) :**

**nextR, nextC = now[0]+dr[i],now[1]+dc[i]**

**if 0<=nextR<n and 0<=nextC<m and visited[nextR][nextC]==0 and board[nextR][nextC]==1 :**

**q = [(nextR,nextC,now[2]+1)]+q**

**visited[nextR][nextC]=now[2]+1**

**print(visited[n-1][m-1])**