**[선형구조: 스택, 큐]**

**선형구조:**

**-** 자료를 저장하고 꺼내는 것에 목적

**스택**: 선입후출 🡺 리스트로 구현 가능

1. *배열*로 접근:

- arr[2]처럼 바로 접근 가능하지만 시작부터 데이터의 최대 길이를 정해야 함

- 데이터 삽입과 삭제가 매우 비효율적

- 데이터의 양이 많고 삽입과 삭제가 잘 일어나지 않으며 데이터 접근이 빈번할 때 유용

2. *연결 리스트로* 접근:

- 데이터의 최대 개수가 정해져 있지 않음

- 데이터의 삽입과 삭제가 효율적

- 데이터들이 순차적으로 있지 않다

- 연결 리스트를 구성하는 요소들 = *노드*

- 한 번에 데이터로 연결 불가능, 주소값을 찾아서 연결하는 방식 때문에

- 삽입 삭제가 빈번하고 접근이 거의 없을 때 유용

큐: 선입선출 🡺 효율성을 위해 데크(deque)로 구현



1. *Enqueue*: 큐 맨 뒤(Insertion 위치)에 요소를 추가

2. *Dequeue*: 큐 맨 앞(Deletion 위치)에 요소를 삭제

3. *Peek*: Front에 위치한 데이터를 읽음

4. *Front*: 큐의 맨 앞 데이터의 인덱스

5. *Rear*: 큐의 맨 뒤 데이터의 인덱스

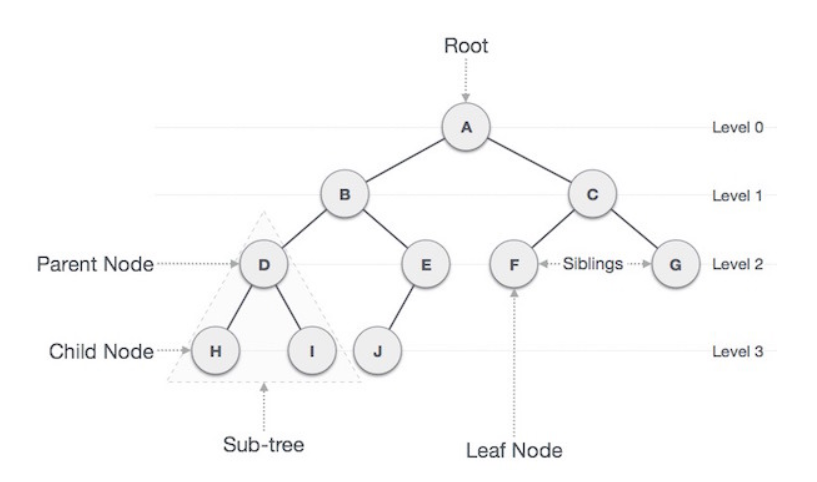
🡺 배열과 연결리스트로 큐를 구현할 수 있음

**[비선형 구조: 트리]**

**비선형 구조**

- 데이터가 계층적으로, 혹은 망으로 구성되어 있음

- 데이터의 표현에 초점을 맞춤



*Root Node*: 트리 구조에서 최상위(A)

*Node*: 트리의 구성요소(A, B, C, D…)

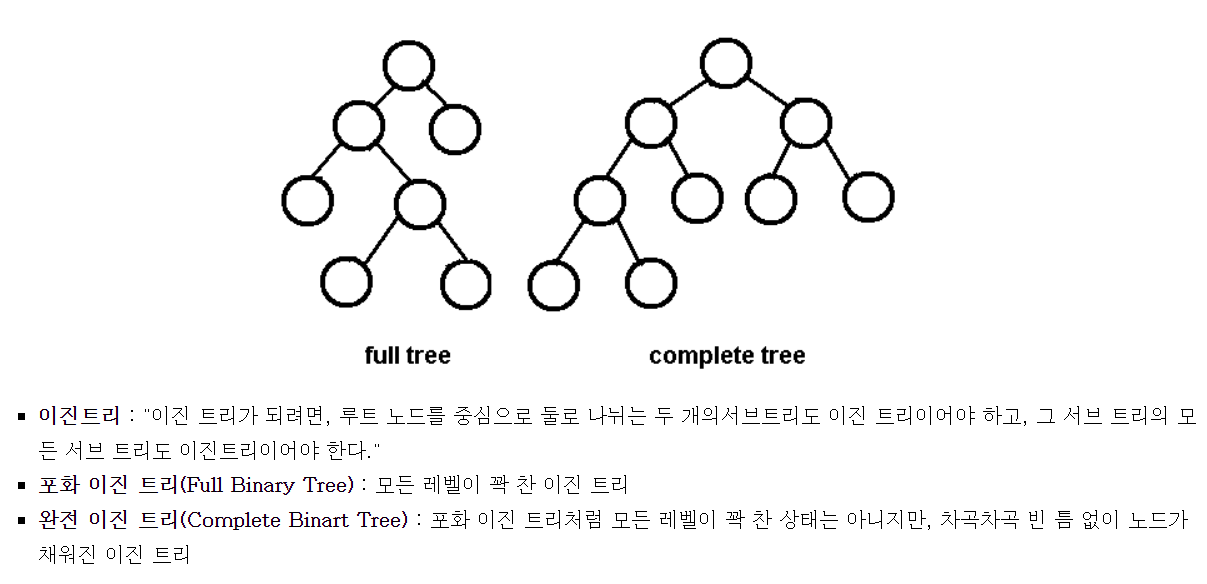
*Edge:* 노드와 노드를 연결하는 연결선

*Terminal Node(Leaf Node)*: 밑으로 노드가 연결되지 않은 노드(H, I, J, F, G)

*Sub-Tree*: 큰 트리에 속하는 작은 트리

*Level*: 트리에서 각 층별로 숫자를 매김

*Height*: 트리의 최고 레벨(3)



🡺 요거는 그냥 참고용

🡺 트리도 배열과 연결리스트 모두 사용 가능

🡺 힙(heap)과 연관된 내용(내일 문제 풀 때 학습)

### 다시 ### 문제 1: 가장 큰 수

이론을 학습하긴 했는데 모르겠어서 일단 강의 시청

나보다 작은 숫자가 내 앞에 있으면 되지 않는다는 생각!

스택은 결국 내림차순으로 정렬되기 때문에 삭제 횟수가 남는다면 맨 뒤부터 제거(stack[:-m])

import sys

sys.stdin=open("input.txt", "r")

n,m = map(int, input().split())

n = list(map(int,str(n)))

stack = []

for x in n:

    while stack and m > 0 and stack[-1] < x:

        stack.pop()

        m-=1

    stack.append(x)

if m != 0:

    stack = stack[:-m]

res = ''.join(map(str, stack))

print(res)

🡺 강의를 들으니까 어느정도 stack에 대한 감이 잡힌다

🡺 마지막 부분은 for x in stack으로 표현해도 됨(아직까지는 실전에서는 이거 쓸 듯)

### 다시 ### 문제 2: 쇠막대기

이것도 정말 모르겠어서 영상 시청

import sys

sys.stdin=open("input.txt", "r")

n = input()

stack = []

cnt = 0

for i in range(len(n)):

    if n[i] == '(':

        stack.append(n[i])

    else:

        if n[i-1] == '(':

            stack.pop()

            cnt+=len(stack)

        else:

            stack.pop()

            cnt+=1

print(cnt)

🡺 이 문제는 이해하는게 좀 많이 어렵다

🡺 이걸 어떻게 생각하지…

🡺 else 이후가 생각하기 매우매우 어려웠다

🡺 stack을 이런 식으로 활용할 수 있구나 싶은 문제

### 다시 ### 문제 3: 후위표기식 만들기

후위표기식:

중위 표기법으로 표기된 3+2+4\*5+3/1를 후위 표기법으로 바꿔본다.

우선 연산자 우선순위에 맞게 괄호를 쳐준다.

((3+2)+(4\*5))+(3/1)

이 괄호 안에 있는 연산자들을 뒤로 빼준다.

(((32)+(45)\*)+(31)/)+

괄호를 모두 없애준다.

32+45\*+31/+

해당 식이 후위 표기법으로 나타낸 식이다.

(<https://simsim231.tistory.com/54> 참고)

**[규칙]**

**1. 숫자가 나오면 그대로 출력한다.**

**2. (나오면 스택에 push한다.**

**3. \* / 나오면 스택에 push한다.**

**4. + - 연산이 나오면 여는 괄호('('), 여는 괄호가 없다면 스택의 끝까지 출력하고 그 연산자를 스택에 push한다.**

**5. 닫는 괄호(')')가 나오면 여는 괄호('(')가 나올때까지 pop하여 출력한다.**

🡺 나 이건 진짜로 이해 코드를 봐도 이해 못했음

🡺 강의 들으면서 이해 시작

1. +와 -는 가장 늦게 연산이 되기 때문에 나온다면 stack에 쌓인 연산자들 다 빼내야 한다(스택이 비어있지 않다면)

2. \*와 /의 경우 stack에 \*와 /가 있지 않은 경우에는 연산자들을 빼내고 stack에 append

3. 여는 괄호는 무조건 append

4. ( 뒤에 +와 -: 여는 괄호 다음에것들 다 빼내야 함

5. 닫는 괄호 나오면 괄호 안에꺼 다 빼낸다

import sys

#sys.stdin=open("input.txt", "r")

n = input()

stack = []

con = ''

for x in n:

    if x.isnumeric():

        con += x

    else:

        if x == '(':

            stack.append(x)

        elif x == '\*' or x == '/':

            while stack and (stack[-1] == '\*' or stack[-1] == '/'):

                con += stack.pop()

            stack.append(x)

        elif x == '+' or x == '-':

            while stack and stack[-1] != '(':

                con += stack.pop()

            stack.append(x)

        elif x == ')':

            while stack and stack[-1] != '(':

                con += stack.pop()

            stack.pop()

while stack:

    con += stack.pop()

print(con)

🡺 이해하기도 버겁다

🡺 거의 외우다싶이 해야 이해가 간다.

🡺 )부분에서 stack.pop()을 빼먹으면 stack에 ( 남으니까 주의

문제 4: 후위식 연산

🡺 각각 pop을 해도 원래 리스트에서 사라짐(이게 핵심)

import sys

sys.stdin=open("input.txt", "r")

n = input()

stack = []

for x in n:

    if x.isnumeric:

        stack.append(x)

    else:

        if x == '\*':

            b = stack.pop()

            a = stack.pop()

            stack.append(a\*b)

        elif x == '/':

            b = stack.pop()

            a = stack.pop()

            stack.append(a/b)

        elif x == '+':

            b = stack.pop()

            a = stack.pop()

            stack.append(a+b)

        else:

            b = stack.pop()

            a = stack.pop()

            stack.append(a-b)

print(stack)

🡺 그냥 n이 문자열로 분해되서 결과가 나온다. 왜지?

import sys

sys.stdin=open("input.txt", "r")

n = input()

stack = []

for x in n:

    if x.isnumeric():

        stack.append(int(x))

    else:

        if x == '\*':

            b = stack.pop()

            a = stack.pop()

            stack.append(int(a\*b))

        elif x == '/':

            b = stack.pop()

            a = stack.pop()

            stack.append(int(a/b))

        elif x == '+':

            b = stack.pop()

            a = stack.pop()

            stack.append(int(a+b))

        else:

            b = stack.pop()

            a = stack.pop()

            stack.append(int(a-b))

print(stack[0])

🡺 숫자들이면 int형으로 바꿔야 한다.

🡺 x.isnumeric: 이렇게 할 경우 걍 다 numeric으로 처리 -> x.isnumeric()으로 꼭!

🡺 마지막에 리스트로 표기되는거 주의

문제 5: 공주 구하기(큐 자료구조로 해결)

deque 참고: <https://leonkong.cc/posts/python-deque.html>

import sys

sys.stdin=open("input.txt", "r")

n,k = map(int, input().split())

a = list(range(1,n+1))

from collections import deque

a = deque(a)

while len(a)>= k:

    a.remove(a[k-1])

    a.rotate(-2)

    if len(a) == 1:

        print(a[0])

🡺 이렇게 하니까 결과가 안나온다. 뭐가 문제지?

import sys

sys.stdin=open("input.txt", "r")

n,k = map(int, input().split())

a = list(range(1,n+1))

from collections import deque

a = deque(a)

while len(a)>=1:

    a.rotate(-2)

    a.popleft()

    if len(a) == 1:

        print(a[0])

🡺 조금 더 단순하게 생각했더니 답 나온다

🡺 굳이 제거하고 rotate할 필요가 없다

🡺 3번부터 오답 나온다 왜지?

import sys

#sys.stdin=open("input.txt", "r")

n,k = map(int, input().split())

a = list(range(1,n+1))

from collections import deque

a = deque(a)

while len(a)>=1:

    a.rotate(-(k-1))

    a.popleft()

    if len(a) == 1:

        print(a[0])

🡺 rotate 숫자를 변수지정 안했었다.

문제 6: 응급실

import sys

#sys.stdin=open("input.txt", "r")

n,m = map(int, input().split())

dan = list(map(int, input().split()))

from collections import deque

dan = deque(dan)

seq = deque(list(range(1,n+1)))

for i in range(n):

    if dan[0] < max(dan):

        dan.rotate(-1)

        seq.rotate(-1)

    else:

        break

seq = list(seq)

print(seq.index(m+1))

🡺 1번과 4번만 맞았다 왜지?

🡺 문제를 잘못 이해함. max면 빼고 다시 로테이트 해야함

import sys

sys.stdin=open("input.txt", "r")

n,m = map(int, input().split())

dan = list(map(int, input().split()))

from collections import deque

dan = deque(dan)

seq = deque(list(range(n)))

cnt = 0

while True:

    num = dan.popleft()

    seq1 = seq.popleft()

    if num != max(dan):

        dan.append(num)

        seq.append(seq1)

    elif num == max(dan):

        cnt+=1

        if seq1 == m:

            break

print(cnt)

🡺 무한루프가 돌아간다…

import sys

sys.stdin=open("input.txt", "r")

n,m = map(int, input().split())

dan = list(map(int, input().split()))

from collections import deque

dan = deque(dan)

seq = deque(list(range(n)))

cnt = 0

while True:

    num = int(dan.popleft())

    seq1 = int(seq.popleft())

    if num < max(dan):

        dan.append(num)

        seq.append(seq1)

    elif num >= max(dan):

        cnt+=1

        if seq1 == m:

            break

print(cnt)

🡺 정답은 맞췄는데 왜 방금껀 안되고 이건 되는지 모르겠다.