

- next\_permu 구하는거 직접 짜고 연습해라 : 재귀로 쓰세요

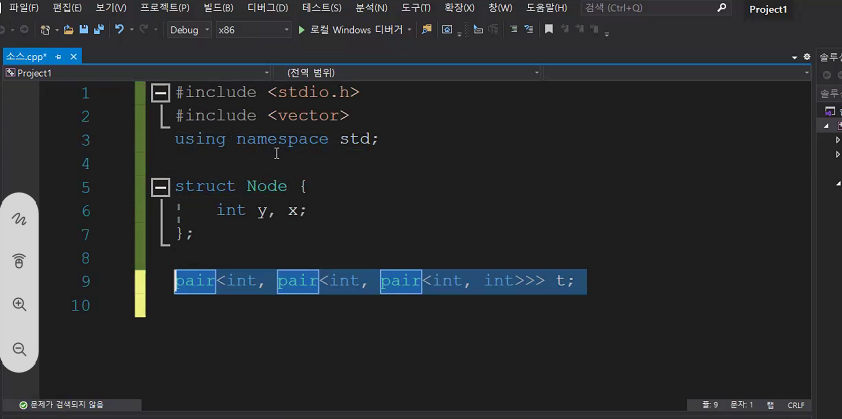
- vector

- 우선순위 큐

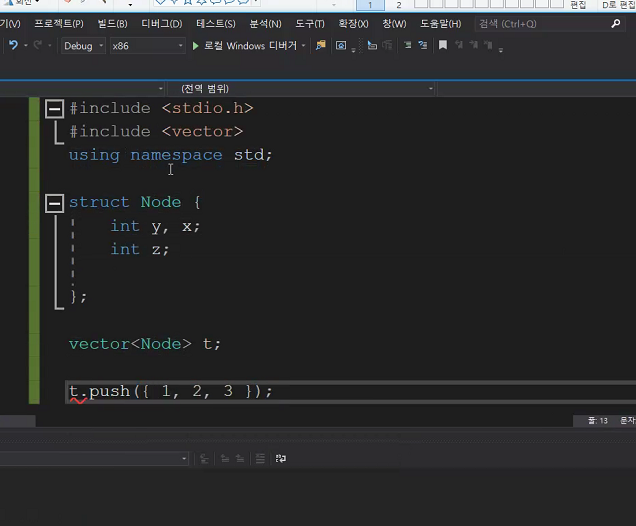
- 너무 많은 STL은 독이 됩니다

- make\_pair 쓰지마세요!! -> 구조체 쓰세요, 사용법은 알고있어야해요, 프로그래머스에서 스타트코드로 주니까

- 페어되신 구조체쓰세요(BFS/DFS 쓸떄 xy좌표



차라리 구조체를 쓰고 이렇게 쓰세요



페어를 쓸 이유가 별로 없어요!! 다 구조체로 쓰면 됩니다

- 정렬(소트)도 직접 구현하세요

- 단순한 플러드필 돌리는 문제는 잘 안나와요

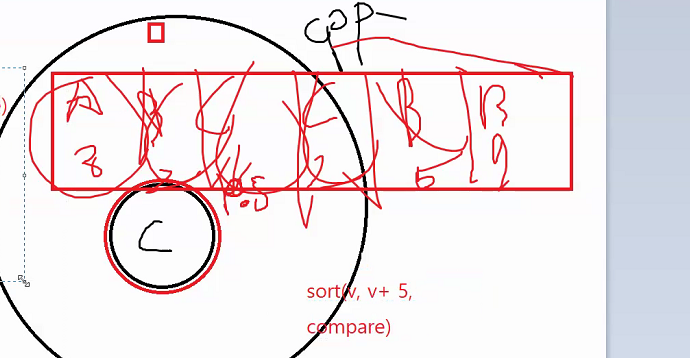
- 페어 쓰다가 런타임 에러 나면 메시지보고 해결못해 그래서 페어는 추천하지않음, 강제는 아니지만 경험자의 조언이다!

- 정렬에서는 컴페어 함수 만들어서 쓰세요.

: sort(start, end, compfunc)

- 내림차순, 오름차순, 다중조건:

- 다중조건 : 노드 구조체 배열 만들어서



2분컷으로 해서 연습해주세요

프라이어리티 큐 (우선순위큐 힙)

- 컴페어 연습 (1오름차순/2내림차순/3다중조건) : 2분컷으로 연습해서 언제든지 쓸 수 있도록

- 좋은 훈련법 :

- 다익스트라 2분컷

- 스루스칼 MST

- 순열

- 조합

###### 다 연습하는데 하루면 된다 하루가 효과가 크다ㅑ!!!

- 디버깅 방법에 대하여

1. 뇌버깅/매의눈   
2. print

3. Trace

OOP 는 C++부터 시작

C++는

면접에서 OOP 설명 교과서적X

|  |
| --- |
| 먼저 구조체가 있었어요 나만의 타입을 만든거야 abcd  나만의 타입은 abcd t  클래스 = 구조체+함수      여기서 중요한 특징은 소스코드도 넣을 수 있다! |
| 파이썬에서는  class KFC():  a = 10  b = 20  def go(self):  print("##$$")   def ago():  print("aaaa")  t=KFC() t.go() t.ago() print(t.a) print(t.b) |
| 어트리뷰트 = 맴버함수, 맴버변수  맴버함수 = 매서드  어트리뷰트 = 맴버변수  인스턴스:  매서드 :  어트리뷰트 :  KFC라는 클래스에서 어트리뷰트는 a,b  OOP를 하면 복사 붙여넣기 해서 버그도없이 그냥쓸수 있다 : 재사용성이 좋다 , 진짜 재사용한다  남들이 짠 소스코드 가져다가 따운받아서 코딩이 가능(블록식 코딩)  객체 단위로 설계한 것을 OOP라고 한다  OOP에서 중요한 것은 재사용성을 염두 해 두고 해야한다.  클래스 제작자/클래스 사용자 =>> 서버코드/클라이언트 코드  코드를 누가써도 안전하게끔 다 막아줘야되  사람들은 프로그램의 설명서를 아무도 읽지 않는다.      \_ \_ ingi ->> 이렇게 하면 사용자들이 마음대로 이상하게 사용하는걸 막을 수 있다.  그래서 세터와 게터를 만들어서 구현해랏. |
| 1번문제  두개의 문자열을 입력 -> 시스템상 없음 채점오류  가장 긴 문자열을 찾는 문제 : BCDE , 유명한 DP문자임, 하지만 여기선 for문과 문자열파싱 |
| Day3 최종프로젝트  - 세부적인 기능들은 OS에서 라이브러리로 편하게 미리 만들어놓고 함수로 제공  - 윈도우의 경우 Win32API  - a[‘list’][i][‘di’] |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |