

온라인 코딩테스트 안내

- 주어진 문제를 읽고 알맞은 프로그램을 작성하여 제출하면 됩니다.
- 완성하신 프로그램에 대한 간단한 설명과 Computational Complexity (Time, Space and others if necessary) 를 주석 형태로 코드에 넣어서 제출해주세요.
 - 간단하게, 이 프로그램이 주어진 입력에 따라서 어느정도 연산을 수행할지, 메모리를 소모할지 분석해주세요. 대략적인 감으로 제시해주셔도 괜찮습니다, 정확하게는 Big O notation (점근 표기법(asymptotic analysis) 중 하나)으로 제시해주면 가장 좋습니다.
 - 예를 들어 Quick Sort는 N 개의 객체를 정렬할때 평균적으로 $O(N\log N)$, 최악의 경우 $O(N^2)$ 번의 비교연산을 수행합니다.
- 채점 기준
 - 코드의 정확성
 - 주어진 조건하에서 오답이 나오면 안됩니다.
 - 문제에 주어진 예시 입출력을 만족시킨다 하여 정확한 코드임을 보장할 수는 없습니다. 가능한 다양한 입력값에 테스트를 해보는 것을 권장합니다.
 - 코드의 효율성
 - Computational Complexity를 고려하여 적절한 구현 방식을 채택해야 합니다.
 - 구동되는 컴퓨터는 일반적인 PC 수준으로 가정
 - 효율성과 정확성이 충돌할 때 정확도를 우선으로 생각해주십시오.
 - 정확성을 희생하고 효율을 얻는 방식보다는 항상 옳은 답안을 내는 구현을 우선시 해주세요.
 - 코드의 가독성
 - 주석도 좋지만 코드 자체로 가독성이 좋은 코드를 더 선호합니다.
- 지원 언어
 - C / C++
 - Java
 - Python
 - Javascript (Node.js)

- 컴파일러 및 입출력 형식 안내
 - 본인의 개발 환경을 간단하게 같이 제출해주세요
 - 정상동작 하지 않아도 코드 상 나타나는 logic이 분명히 맞다면
정답으로 간주할 수 있으나, 정확한 개발환경을 명시해서 프로그램을
구동해볼 수 있도록 해주세요.
 - 개발 환경 예시
 - Linux, g++ 7.3.0
 - Windows, Python 3.7.0
 - macOS, node.js v8.12.0
 - ...
 - 입출력은 표준 입출력 (standard input, output)을 사용합니다.
 - 주 언어가 Javascript의 경우 Node.js를 사용할 수 있습니다.
 - 표준 입출력으로 동작해야 완성된 답안으로 인정받을 수 있지만 시간이
모자라거나 표준 입출력에 익숙하지 않은 경우 어떤 방식으로든
입출력이 가능하게 만들어주세요.
 - 예를 들어, 입력값을 코드에 직접 넣는다면 그 부분을 주석으로
표시해서 테스트가 가능한 형태로 만들어주세요.
- 시간이 모자라면 모든 문제에 애매하게 손을 대기 보다는 한 문제라도 정확한
구현체를 작성하고 나머지 문제는 아이디어를 제시하는 식으로 해주시는게 좋습니다.
 - 구현을 마무리하지 못한 문제는 pseudo-code 혹은 풀이 아이디어를 주석으로
코드 안에 포함해서 제출해주세요
- 각 코드 파일 이름은 문제 번호.cpp, py, js 등으로 만들고 한 번에 압축하여
보내주세요 (압축을 안하면 첨부파일로 전달이 안되는 경우가 생길 수 있습니다.)
 - 문제 풀이와 관련된 내용은 메일 본문보다는 모두 코드 안에 포함시켜주세요
(자연어로 된 풀이 설명 등 역시 코드 안에 주석으로 넣어주세요.)
- 시간 : 90분