

# 알고리즘 문제해결전략

2018102182 김지수



# CONTENTS

- 01. 문제 해결과 프로그래밍 대회
- 02. 문제 해결 개관
- 03. 코딩과 디버깅에 관하여

# 01

## 문제 해결과 프로그래밍 대회

문제 해결 기술을 연마하기에  
가장 좋은 방법 !

- ACM-ICPC(ACM 대학생 프로그래밍 경시대회)
- 탑코더(TopCoder)
- 구글 코드 잼(Google Code Jam)

- 가능한 한 많은 문제 풀기
- 온라인 채점 사이트 이용
- 많은 프로그래밍 대회 참가

- 팀 단위 연습을 위한 팁
  - 1) 종이 위에 답안 스케치하기
  - 2) 역할 분담
  - 3) 페어 프로그래밍 연습하기
  - 4) 디버거 없이 디버깅
  - 5) 두 화면 분할

# 02

## 문제 해결 개관



1. 문제 읽고 이해하기
2. 익숙한 용어로 재정의
3. 어떻게 해결할지 계획 세우기
4. 계획 검증하기
5. 프로그램으로 구현하기
6. 어떻게 풀었는가 돌아보고, 개선 방법 찾기

# 03

## 코딩과 디버깅에 관하여

- 간결한 코드 작성
- 적극적인 코드 재사용
- 표준 라이브러리 공부
- 항상 같은 형태로 프로그램 작성
- 일관적이고 명료한 명명법 사용 : 모호한, 변수or함수명 X
- 모든 자료를 정규화해서 저장
- 코드와 데이터 분리

- 산술 오버플로
- 배열 범위 밖 원소 접근
- 일관되지 않은 범위 표현 방식 사용
- Off-by-one 오류 : 하나가 모자라거나 많아서 나는 오류
- 상수 오타
- 스택 오버플로
- 다차원 배열 인덱스 순서 바꿔 쓰기
- 너무 느린 입출력 방식 선택
- 변수 초기화

- **디버깅** : 프로그래밍 대회에서는 유용성이 제한
    - 1) 눈으로 하는 디버깅이 빠른 경우가 많음
    - 2) 복잡한 코드는 디버깅 X
- > 디버거 없이 프로그램 버그 찾아내는 연습

- 1) 작은 입력에 대해 제대로 실행되나 확인
- 2) 단정문 쓰기 : 거짓일 때 강제 종료시키는 구문
- 3) 프로그램 계산 중간 결과를 출력

이래도 모르겠다면 ... 디버거 사용 ...

## 산술 오버플로?

- 컴퓨터의 모든 변수에는 담을 수 있는 크기 제한
- 예상과 다르게 동작하는 경우 발생
  - 너무 큰 결과
  - 너무 큰 중간값
  - 너무 큰 '무한대' 값

## 오버플로 피해가기 !

- 더 큰 자료형 쓰기
  - 연산의 순서 바꾸기
    - 자료형의 **프로모션** : 피연산자의 자료형이 다르거나 범위 작은 경우, 컴파일러가 변환해 계산
- 1) 하나 정수, 하나 실수 -> 실수형
  - 2) 둘 다 정수, 둘 다 실수 -> 더 넓은 범위 갖는 자료형
  - 3) 둘 다 int형보다 작은 정수 -> 둘 다 int형
  - 4) 부호 없는 정수형, 부호 있는 정수형 혼합 -> 부호 없는 정수형



컴퓨터의 모든 실수 변수는 정확도가 제한된 **근사값** 저장

-> 실수 계산에서는 작은 환경에 따라서 **값이 변할 수 있음**

**IEEE 754 표준** : 가장 많이 사용되는 실수 표기 방식

- 실수를 이진법으로 표기
- 부동 소수점 표기

소수점 위치 풀이

유효숫자

자료형	부호 비트	지수 비트	가수 비트
32비트 실수형	1	8	23
64비트 실수형	1	11	52
80비트 실수형	1	15	64

#### 1) 오차 한도 정하기

같다고 판단해야 하는 두 개 비교

-> 가능한 한 최대치 구한 뒤, 더 큰 오차 한도 값 사용

다르게 판단해야 하는 두 개 비교

-> 하한을 계산하면 상한 값 정할 수 있음

#### 2) 상대 오차 이용하기

비교하는 숫자들의 크기에 비례해 오차 정하는 방식

#### 실수 연산 하지 않기 !

- 곱셈과 나눗셈 순서 바꾸기
- 양변 제공하기
- 기하 문제에서 좌표계를 정수배 늘리기

프로그래밍 대회에서는  
작은 오차가 있어도 **정답 인정 ^^!**

# THANK YOU

Presentation Design Group **D'Breed**

[www.dbreed.co.kr](http://www.dbreed.co.kr)

*D'breed*