

♥ 정적 분석 (Static Analysis) 개요

☑ 정의

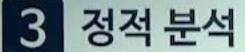
■ 자동화된 도구 등의 지원을 받아 정적 테스트를 수행하는 것

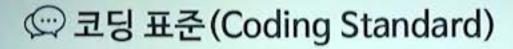
SEJONG CYBER UNIVERSITY

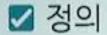
☑ 종류

- 코딩 표준
- 코드 복잡도 계산
- 자료 흐름 분석









■ 개발자가 프로그램을 작성할 때 지켜야 하는 규약

[코딩표준 적용 전]

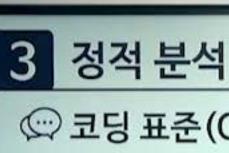
int foo(int n) { int i=0; while (n)0) {i = i + n % 10; n = n/10; } return i;

[코딩표준 적용 후]

```
public static int foo (int n) {
  int i = 0;
  while (n > 0) {
    i = i + n % 10;
    n = n / 10;
  }
  return i;
}
```







- 코딩 표준(Coding Standard)
 - ✓ MISRA-C
 - 대표적인 코딩 표준
 - 영국 자동차 산업 신뢰성 협회(Motor Industry Software Reliability Association)에서 발표한 C 프로그래밍 언어의 가이드라인

SEJONG CYBER UNIVERSITY

- 안전하고, 이식성 좋고, 신뢰성 있는 코드
- ☑ 코딩 규칙 검사 도구 예시
 - LDRA
 - PRQA 등



3 정적 분석

의 복잡도 분석(Complexity Analysis)

☑ 개요

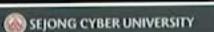
- 복잡도가 필요 이상으로 높지 않도록 통제 필요
- 순환 복잡도(Cyclomatic complexity)가 가장 널리 사용

🙆 세종사이버대학교

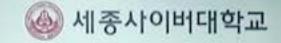
학습하기

☑ 순환 복잡도

- 주어진 제어 흐름 그래프에서 선형적으로 독립적인 기본 경로(Basis path)라 불리는 프로그램 경로들의 개수
- E: 제어 흐름 그래프에 있는 간선 개수
- N: 노드 개수 순환 복잡도 = E N + 2 = 닫힌 영역의 개수 + 1 = 분기 노드 개



3 정적 분석



학습하기

의 복잡도 분석(Complexity Analysis)

☑ 개요

- 복잡도가 필요 이상으로 높지 않도록 통제 필요
- 순환 복잡도(Cyclomatic complexity)가 가장 널리 사용

☑ 순환 복잡도

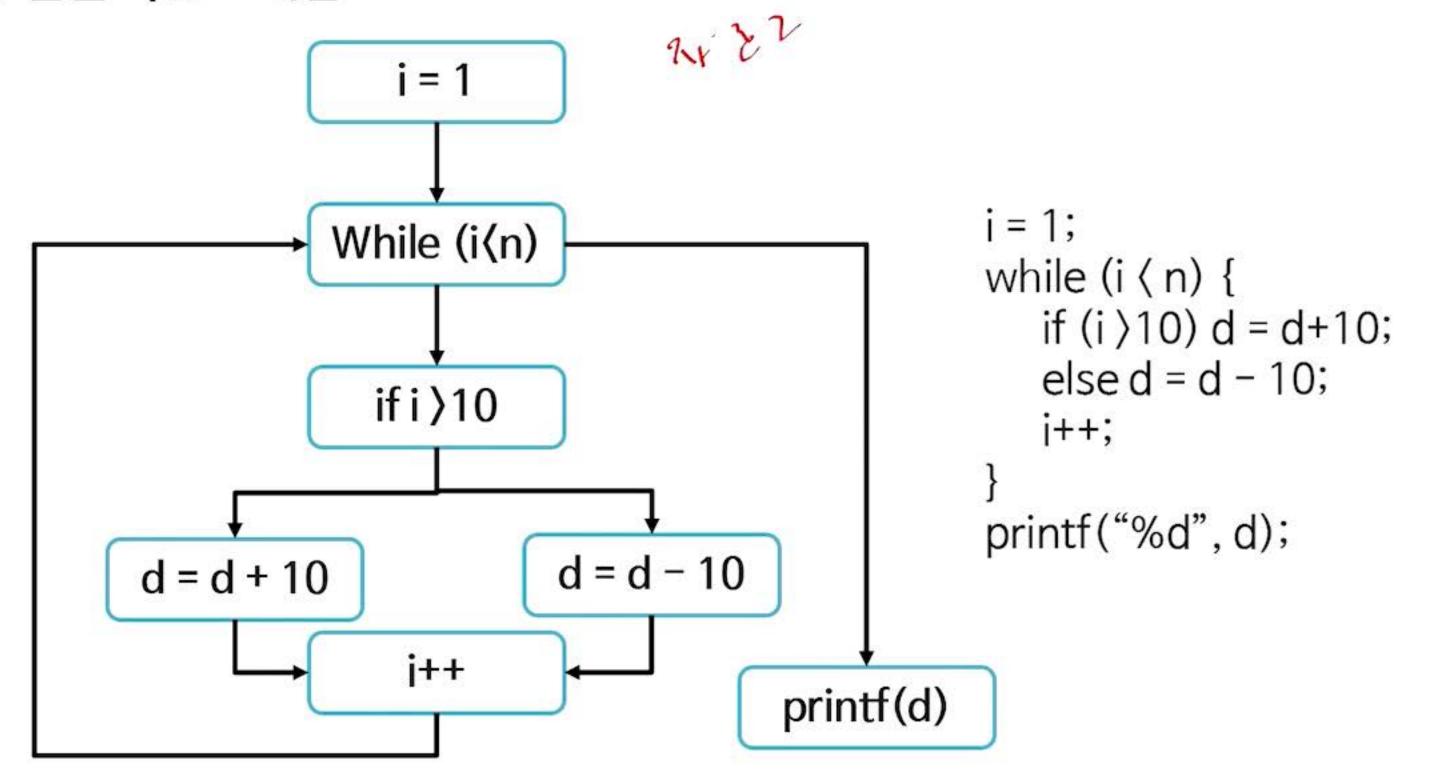
- 주어진 제어 흐름 그래프에서 선형적으로 독립적인 기본 경로(Basis path)라 불리는 프로그램 경로들의 개수
- E: 제어 흐름 그래프에 있는 간선 개수
- N: 노드 개수 순환 복잡도 = E N + 2 = 닫힌 영역의 개수 + 1 = 분기 노드 개수 + 1





☞ 순환 복잡도 계산



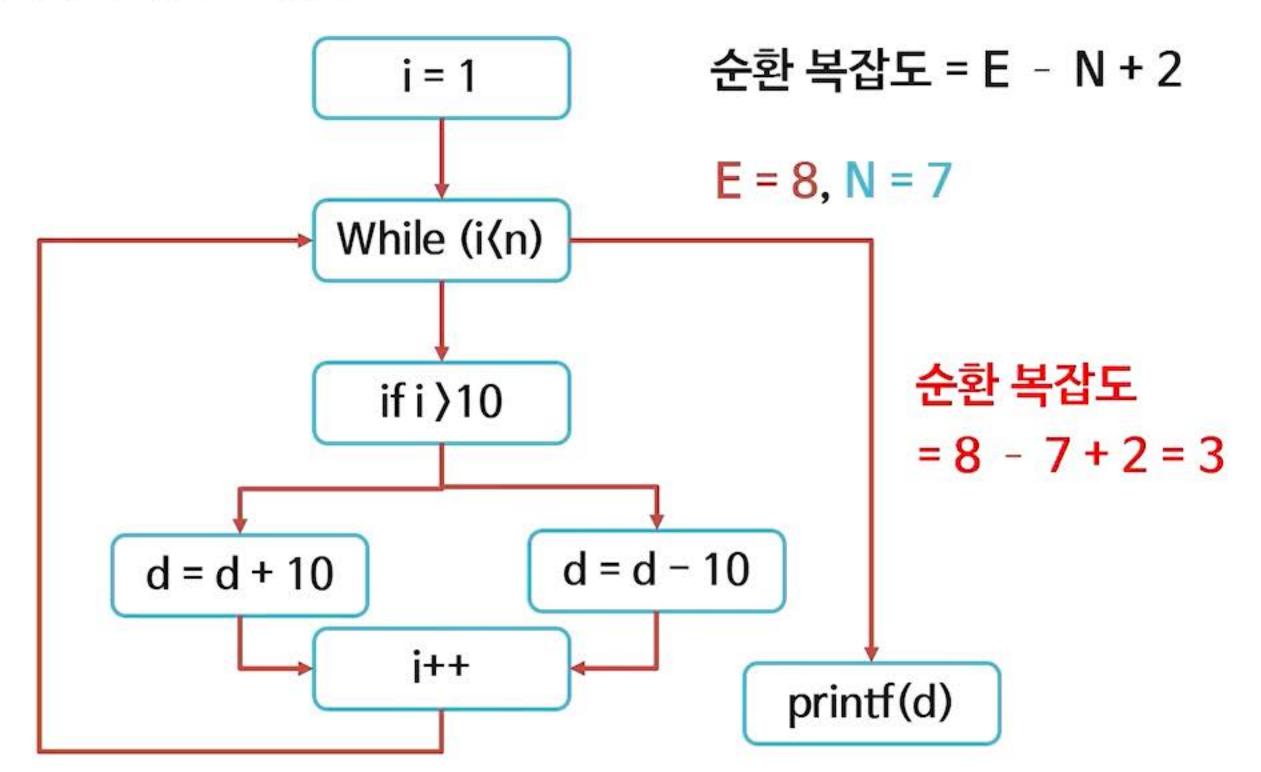






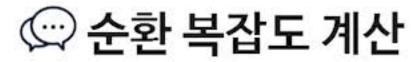
☞ 순환 복잡도 계산

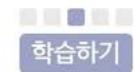


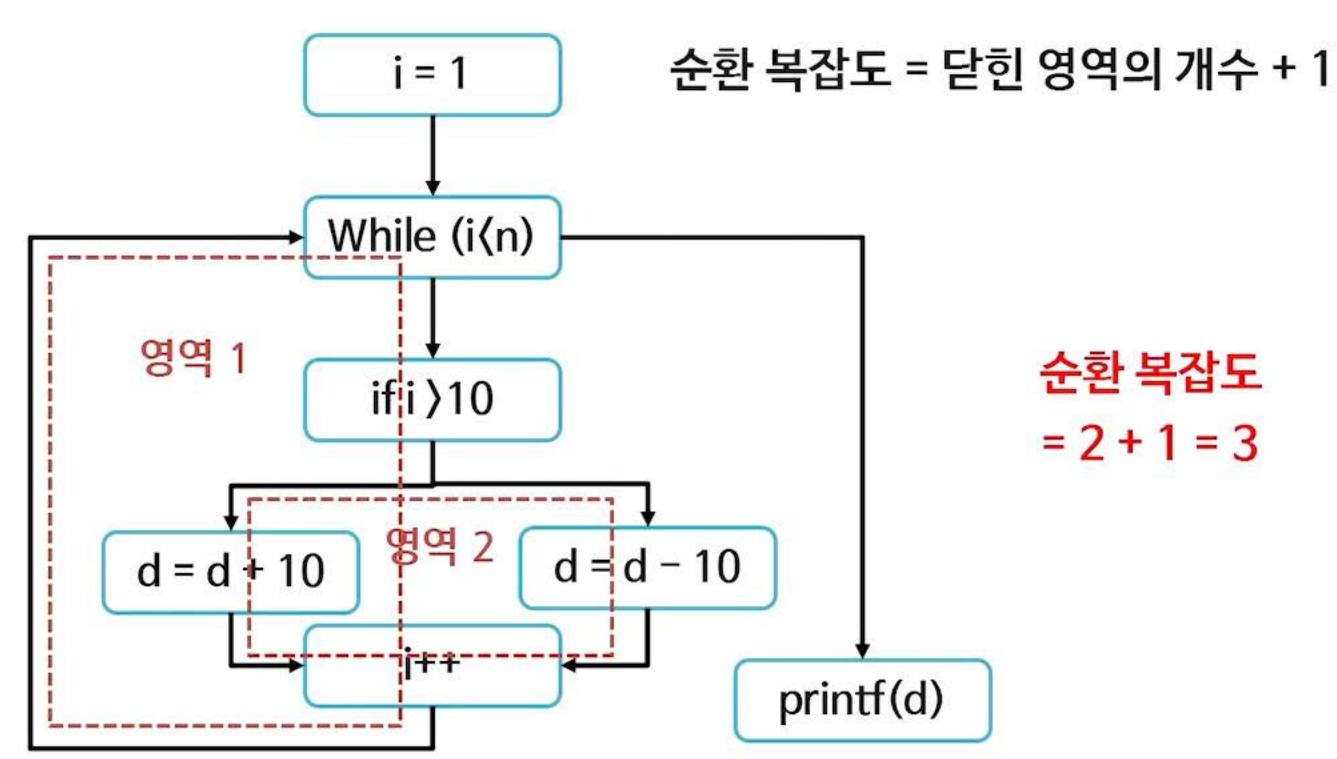




4 1 5





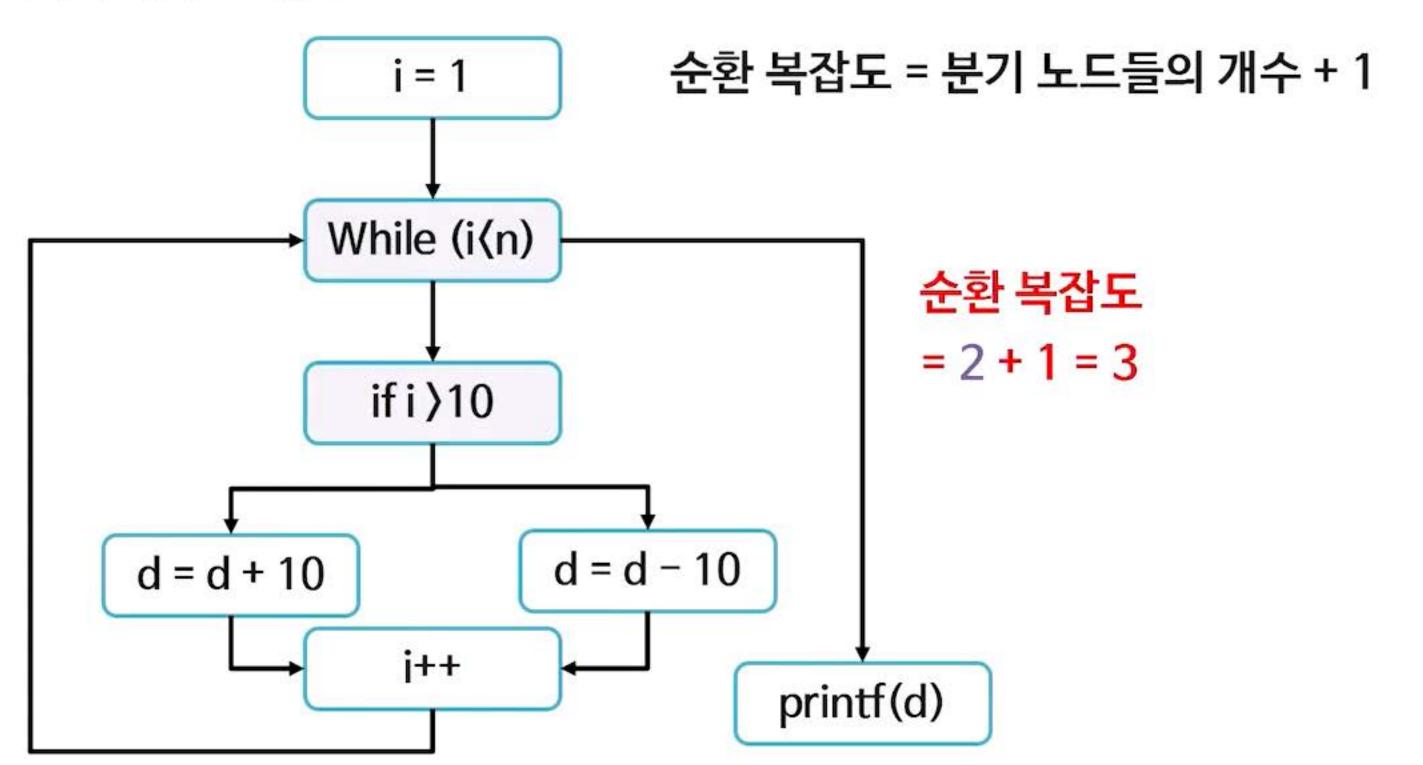




1 1 5



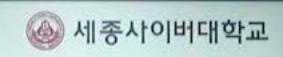






2 1 5

정적 분석



학습하기

☞ 자료 흐름 분석 예제

1: int dataflow (int c, int d) {

2: int a=0, x:

a = c+d;

if (a)0

x = 1; 2 else

c = 10;6:

a = a+x00

8: return a;

9: }

[변수 x에 대한 자료 흐름 패턴]

| 자료 흐름 패턴 | | 설명 |
|----------|-----------|----------------------------------|
| du | 5-7 | 문제없음. 정상 |
| Qu | 2-3-4-6-7 | 잠재적 결함, 자료가 정의되지 않고 바로 사용됨 |
| k∼ | 9) - | 문제없음. 정상 |

