수학적 귀납법, 루프 불변

연세대 미래캠퍼스 고급 알고리즘 강의 1일차

이혜아

2022년 7월 6일

수학적 귀납법

수학적 귀납법 (Mathematical Induction)

어떤 명제 $P(n)$이

1. $n = 1$일 때 참이고,
2. $n = k$일 때 참이면, $n = k + 1$일 때도 참

이면, $P(n)$은 모든 $n \geq 1$인 정수에 대해서 참이다.

수학적 귀납법의 다른 형태

강한 수학적 귀납법 (Strong Mathematical Induction)

어떤 명제 $P(n)$이

1. $n = 1$일 때 참이고,
2. $n \leq k$일 때 참이면, $n = k + 1$일 때도 참

이면, $P(n)$은 모든 $n \geq 1$인 정수에 대해서 참이다.

자연수의 정렬성 (Well-ordering principle)

공집합이 아닌 임의의 자연수의 부분집합은 최소원소를 가진다.

$$\forall X \subseteq \mathbb{N}, X \neq \phi \Rightarrow \exists x : \forall y \in X, x \leq y$$

자연수의 정렬성 $\Rightarrow$ 수학적 귀납법

자연수의 정렬성 $\Rightarrow$ 수학적 귀납법

자연수의 정렬성을 이용하여 수학적 귀납법을 증명할 수 있다.

- 귀류법을 사용하자.
- 귀납법 성질을 만족하는 $P(n)$이 거짓인 n이 존재한다고 하자.
- $\phi \neq \{n \mid P(n)\text{이 거짓}\}$ 에는 최소 원소 x가 있다.
  - $x = 1$이라면, $P(1)$이 성립한다는 것에 모순
  - $x > 1$이라면, $P(x-1)$은 참이고, $P(x-1) \Rightarrow P(x)$에 모순

연습문제

1. *(Hard)* 수학적 귀납법 $\Rightarrow$ 자연수의 정렬성
2. *(Hard)* 수학적 귀납법 $\Rightarrow$ 강한 수학적 귀납법

예시

가우스 합공식

$1+2+\cdots+n=\frac{n(n+1)}{2}$

1. $n=1$: $1=\frac{1(1+1)}{2}$이다.
2. $n=k \Rightarrow n=k+1$ ($k+1$인 경우를 k인 경우(귀납 가설)를 이용해 증명)

$$\begin{aligned}\underbrace{1+2+\cdots+k}_{\text{귀납 가설 이용}}+(k+1) &= \frac{k(k+1)}{2}+\frac{2(k+1)}{2} \\ &= \frac{(k+2)(k+1)}{2} \\ &= \frac{(k+1)\left((k+1)+1\right)}{2}\end{aligned}$$

팁

- 변수가 여럿 있을 때, n은 자유롭게 골라도 된다.
- 수학적 귀납법은 n이 작은 경우를 **보조정리**로 사용하여 큰 경우를 해결하는 것이다.
- 수학적 귀납법과 "조건을 만족하지 않는 최소 반례"를 찾는 것은 같다.

연습문제

1. *(Medium)* $1^3 + 2^3 + \cdots + n^3 = (1 + 2 + \cdots + n)^2$
2. *(Hard)* $2^n \times 2^n$ 격자에서 임의의 칸을 하나 뺀 나머지 부분을, 3칸을 차지하는 ㄱ자 모양의 타일을 이용해 겹치지 않게 덮을 수 있다.
    - 관련 문제: `https://www.acmicpc.net/problem/14601`
3. *(Hard)* 트리의 정점의 수는 간선의 수보다 하나 많다.
    - 증명 과정에서 **이미 알고있는** 트리의 성질을 증명 없이 이용하지 않도록 한다.

루프 불변

max의 정당성

배열 $A_{0 \ldots N-1}$이 주어질 때, 다음 코드는 올바르게 배열 A의 최댓값을 구한다. ($N \geq 1$)

1. $M \leftarrow A_0; i \leftarrow 1;$
2. **while** $i \neq N$ **do**
   1. **if** $M < A_i$ **then** $M \leftarrow A_i$ **end if;**
   2. $i \leftarrow i + 1$

   end while;

while의 성질

- 다음과 같은 **while**의 성질이 있다.
  - 임의로 루프를 탈출하는 **goto**나 **break**는 없다고 한다.
- **while** <조건> **do** <문장> **end while;** 이 있다고 하자.
- **while**문을 빠져나간 이후에 다음이 성립한다.
  - **while**문을 빠져나가지 못하고 무한루프를 돌 수도 있다.

1. 루프 탈출: <조건>을 만족하지 않는다.
2. 루프 불변: 아래 조건에서 <**루프 불변**>이 성립한다.
   1. **while**문이 시작하기 전에 <루프 불변>이 성립한다.
   2. <루프 불변>을 만족하는 상태에서 <문장>을 실행 하면, 실행 이후에 <루프 불변>을 만족한다.

- 수학적 귀납법과 유사한 형태를 띈다.

루프 불변

1. $M \leftarrow A_0; i \leftarrow 1;$
2. **while** $i \neq N$ **do**
   1. **if** $M < A_i$ **then** $M \leftarrow A_i$ **end if;**
   2. $i \leftarrow i + 1$

   end while;

- 루프 불변: M은 $\{A_0, A_1, \cdots, A_{i-1}\}$의 최댓값이다.
  1. 루프를 시작하기 전: M은 $\{A_0\}$의 최댓값이다.
  2. 2-1 시행 이후: M은 $\{A_0, A_1, \cdots, A_i\}$의 최댓값이다.
     - $\{A_0, A_1, \cdots, A_{i-1}, A_i\}$의 A_i가 아닌 최댓값이 있다면 if문을 들어가지 않고, 조건을 만족한다.
     - A_i가 유일한 최댓값이라면, if문을 들어가서 조건을 만족한다.
  3. 2-2 시행 이후: M은 $\{A_0, A_1, \cdots, A_{i-1}\}$의 최댓값이다.
     - i가 1 증가했다.

max의 정당성 – 루프 불변

1. $M \leftarrow A_0; i \leftarrow 1;$
2. **while** $i \neq N$ **do**
    1. **if** $M < A_i$ **then** $M \leftarrow A_i$ **end if;**
    2. $i \leftarrow i + 1$

   end while;

- 루프 불변: M은 $\{A_0, A_1, \cdots, A_{i-1}\}$의 최댓값이다.
- 루프 탈출: $i \neq N$이 성립하지 않는다. $\Rightarrow i = N$이다.
- 두 조건을 합치면, M은 $\{A_0, A_1, \cdots, A_{N-1}\}$의 최댓값이다.
- 루프 불변과 루프 탈출은, 프로그램의 정당성을 부여해준다.

연습문제

연습문제

1. *(Medium)* 선택정렬이 올바름을 루프 불변을 이용해서 증명하여라.
2. *(Hard)* 삽입정렬이 올바름을 루프 불변을 이용해서 증명하여라.
3. *(Hard)* 삽입정렬을 하는 중에 각 단계를 선택정렬을 하도록 하면, 배열은 올바르게 정렬되는가? 그 반대의 경우는 어떻게 되는가?