Today

- Proof by Induction
- Big-Oh
- Algorithm Analysis
Proof by Induction

Base Case:
1. Prove $P(0)$ (sometimes $P(1)$)

Inductive Hypothesis
2. Let $k$ be an arbitrary integer $\geq 0$
   3. Assume that $P(k)$ is true

Inductive Step
4. ...
5. Prove $P(k+1)$ is true
Examples

Solutions

\[ \sum_{i=1}^{n} i = n(n+1)/2 \quad \text{for all } n \geq 1 \]

Solution:

1. Base Case: \( n = 1 \)
   \[ \sum_{i=1}^{1} i = 1 = \frac{1(1+1)}{2} \]

2. Inductive Hypothesis:
   Assume that \( \sum_{i=1}^{k} i = k(k+1)/2 \) is true for all \( k \geq 1 \).

3. Inductive Step: \( (k+1) \)
   \[ \sum_{i=1}^{k+1} i = \sum_{i=1}^{k} i + (k + 1) \]
   \[ = \frac{k(k+1)}{2} + (k + 1) \quad \text{[Inductive hypothesis]} \]
   \[ = \frac{k^2+k}{2} + k + 1 \]
   \[ = \frac{k^2+k}{2} + \frac{2(k+1)}{2} \]
   \[ = \frac{k^2+3k+2}{2} \]
   \[ = \frac{(k+1)(k+2)}{2} \]
   \[ = \frac{2}{(k+1)((k+1)+1)} \]
\[ \sum_{i=1}^{N} i^2 = 1^2 + 2^2 + 3^2 + 4^2 + \cdots = \frac{N(N+1)(2N+1)}{6} \]

**Solution:**

1. **Base Case: n = 1**
   \[ \sum_{i=1}^{1} i^2 = 1^2 = 1 = \frac{6}{6} = \frac{1(1 + 1)(2(1) + 1)}{6} \]

2. **Inductive Hypothesis:**
   Assume that
   \[ \sum_{i=1}^{k} i^2 = \frac{k(k+1)(2k+1)}{6} \]
   for all \( k \geq 1 \).

3. **Inductive Step: (k+1)**
   \[ \sum_{i=1}^{k+1} i^2 = \sum_{i=1}^{k} i^2 + (k + 1)^2 \]
   \[ = \frac{k(k+1)(2k+1)}{6} + (k + 1)^2 \text{ [Inductive Hypothesis]} \]
   \[ = \frac{(2k^3 + 2k^2 + k^2 + k)}{6} + (k + 1)^2 \]
   \[ = \frac{2k^3 + 2k^2 + k^2 + k}{6} + k^2 + 2k + 1 \]
   \[ = \frac{2k^3 + 9k^2 + 13k + 6}{6} \]
   \[ = \frac{(k + 1)(k + 2)(2k + 3)}{6} \]
   \[ = \frac{(k + 1)((k + 1) + 1)(2(k + 1) + 1)}{6} \]

**Examples**

**Solutions**
\[ \sum_{i=1}^{n} \frac{1}{i(i+1)} = \frac{n}{n+1} \quad \text{where} \quad n \in \mathbb{Z}^+ \]

Solution:

Base Case: \( n = 1 \)

\[ \sum_{i=1}^{1} \frac{1}{i(i+1)} = \frac{1}{2} = \frac{1}{1+1} = \frac{n}{n+1} \]

Inductive Hypothesis:

Assume that \( \sum_{i=1}^{k} \frac{1}{i(i+1)} = \frac{k}{k+1} \)

for all \( k \geq 1 \).

Inductive Step: (\( k+1 \)):

We have to prove that

\[ \sum_{i=1}^{n+1} \frac{1}{i(i+1)} = \frac{(n+1)}{(n+1)+1} \]

Taking the left hand side...

\[ \sum_{i=1}^{n+1} \frac{1}{i(i+1)} = \left( \sum_{i=1}^{n} \frac{1}{i(i+1)} \right) + \frac{1}{(n+1)(n+2)} \]

\[= \frac{n}{n+1} + \frac{1}{(n+1)(n+2)} \]

\[= \frac{n^2 + 2n + 1}{(n+1)(n+2)} \]

\[= \frac{(n+1)^2}{(n+1)(n+2)} \]

\[= \frac{n+1}{n+2} \]

\[= \frac{(n+1)}{(n+1)+1} \]

By showing this works for the base case, and then assuming it works for an integer \( n \in \mathbb{Z}^* \), and proving it works for \( n+1 \), we can conclude that it is true for all \( n \in \mathbb{Z}^* \).
Logarithms

• log means log base of 2
• $\log(N^k) = k \log N$
  
  – Eg. $\log(A^2) = \log(A\times A) = \log A + \log A = 2\log A$
Big-Oh

- We only look at worst case
- Big input
- Ignore constant factor and lower order terms
  - Why?
- Definition:

  \[
  g(n) \text{ is in } O(f(n)) \text{ if there exist constants } c \text{ and } n_0 \\
  \text{such that } g(n) \leq c \cdot f(n) \text{ for all } n \geq n_0
  \]

- Also lower bound and tight bound

We use \( O \) on a function \( f(n) \) (for example \( n^2 \)) to mean the set of functions with asymptotic behavior less than or equal to \( f(n) \)
How to analyze the code?

Consecutive statements
  Sum of times
Conditionals
  Time of test plus slower branch
Loops
  Sum of iterations
Calls
  Time of call’s body
Recursion
  Solve recurrence equation
Examples

1. int example1(int n) {
    if (n < 10)
        return n - 1;
    else {
        return example1(n / 2);
    }
}

O(log N)

2. int example2(int n, int sum) {
    for (int k = 0; k < n * n; ++k)
        for (int j = 0; j < k; j++)
            sum++;
    return sum;
}

O(n^4)

3. int example3(int n, int sum) {
    for (int k = n; k > 0; k--)
        for (int i = 0; i < k; i++)
            sum++;
        for (int j = n; j > 0; j--)
            sum++;
    return sum;
}

O(n^2)

(From CSE332 12SP Midterm)
Algorithm Analysis

• What is the Big-Oh for the following?
  – Finding the smallest item in an N-item array
    • Sorted?  \( O(1) \)
    • Unsorted?  \( O(N) \)

  – Hint: What is the worst case location of the item?