Summations Activity - Solutions

Triangular Numbers

Formula: $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

Example 1: Determine the value of the sum 1 + 2 + 3 + ... + 50

Solution: Using the triangular number formula with n = 50: $$\sum_{i=1}^{50} i = \frac{50(50+1)}{2} = \frac{50 \times 51}{2} = \frac{2550}{2} = 1275$$

Answer: 1275

Application: Algorithm Analysis for stockProfit

Solution: Let's analyze the algorithm line by line:

```
stockProfit
n = p.last                    // 1 statement
buy = 1                      // 1 statement
sell = 2                     // 1 statement
profit = p[2] - p[1]         // 1 statement
for i = 1 to n-1             // outer loop runs (n-1) times
    for j = i+1 to n         // inner loop runs variable times (see analysis
below)
        if p[j] - p[i] > profit    // 1 statement per inner iteration
            buy = i                // 1 statement per inner iteration
            sell = j               // 1 statement per inner iteration
            profit = p[j] - p[i]   // 1 statement per inner iteration
println("Buy on day " + buy)       // 1 statement
println("Sell on day " + sell)     // 1 statement
println("Profit: $" + profit)      // 1 statement
```

Analysis:

- Initial statements: 4 statements
- Final print statements: 3 statements
- For the nested loops:
  - When i = 1: j goes from 2 to n, so (n-1) iterations
  - When i = 2: j goes from 3 to n, so (n-2) iterations
  - When i = 3: j goes from 4 to n, so (n-3) iterations
  - ...
  - When i = n-1: j goes from n to n, so 1 iteration

Total inner loop iterations: $(n-1) + (n-2) + (n-3) + ... + 1 = \sum_{i=1}^{n-1} i = \frac{(n-1)n}{2}$

Each inner iteration executes 5 statements (inner for/increment + if condition + 3 assignments).

Outer for loop runs (increments) (n-1) times.

Total statements: $$4 + (n-1) + 5 \times \frac{(n-1)n}{2} + 3 = 7 + (n-1) + \frac{5(n-1)n}{2} = 7 + (n-1)\left(1 + \frac{5n}{2}\right) = 7 + (n-1)\frac{2 + 5n}{2}$$

Simplifying further: $$= 7 + \frac{(n-1)(5n + 2)}{2} = 7 + \frac{5n^2 + 2n - 5n - 2}{2} = 7 + \frac{5n^2 - 3n - 2}{2} = \frac{14 + 5n^2 - 3n - 2}{2} = \frac{5n^2 - 3n + 12}{2}$$

Answer: $\frac{5n^2 - 3n + 12}{2}$ statements

Finite Geometric Sum

Formula: $\sum_{i=0}^{n} ar^i = a \cdot \frac{r^{n+1} - 1}{r - 1}$ (for r ≠ 1)

Example 2: Determine the value of the sum 3 + 6 + 12 + 24 + ... + 3072

Solution: First, identify the pattern:

- First term: a = 3
- Common ratio: r = 2 (since 6/3 = 2, 12/6 = 2, etc.)
- Last term: 3072

Find n by determining when $3 \times 2^n = 3072$: $2^n = 1024 = 2^{10}$, so n = 10

The sum is: $\sum_{i=0}^{10} 3 \times 2^i = 3 \sum_{i=0}^{10} 2^i = 3 \cdot \frac{2^{11} - 1}{2 - 1} = 3 \cdot \frac{2048 - 1}{1} = 3 \times 2047 = 6141$

Answer: 6141

Binary Tree Applications

Example 3: Find the number of nodes in a perfect binary tree of height 12

Solution: In a perfect binary tree:

- Level 0 (root): $2^0 = 1$ node
- Level 1: $2^1 = 2$ nodes
- Level 2: $2^2 = 4$ nodes
- ...
- Level h: 2^h nodes

Total nodes = $\sum_{i=0}^{12} 2^i = \frac{2^{13} - 1}{2 - 1} = 2^{13} - 1 = 8192 - 1 = 8191$

Answer: 8191 nodes

Example 4: Find the range (min and max) for the number of nodes in a complete binary tree of height 6

Solution: For a complete binary tree of height h:

- **Minimum**: All levels filled except the last level has only 1 node
    - Levels 0 through h-1 are completely filled: $\sum_{i=0}^{5} 2^i = 2^6 - 1 = 63$

    - Plus 1 node at level 6: $63 + 1 = 64$ nodes

- **Maximum**: All levels completely filled (this is a perfect tree)

    - $\sum_{i=0}^{6} 2^i = 2^7 - 1 = 127$ nodes

Answer: Range is [64, 127] nodes

Example 5: Find an expression for the number of nodes in a perfect binary tree of height h

Solution: Using the geometric series formula: $$\text{Number of nodes} = \sum_{i=0}^{h} 2^i = \frac{2^{h+1} - 1}{2 - 1} = 2^{h+1} - 1$$

Answer: $2^{h+1} - 1$ nodes

Example 6: Find an expression for the height of a perfect (or complete) binary tree with n nodes

Solution: From Example 5, we know $n = 2^{h+1} - 1$ for a perfect tree. Solving for h: $$n + 1 = 2^{h+1}$$ $$\log_2(n + 1) = h + 1$$ $$h = \log_2(n + 1) - 1$$

For a complete tree, the height is $h = \lfloor \log_2(n) \rfloor$

Answer: $h = \lfloor \log_2(n) \rfloor$ (for complete trees)

Example 7: Find the height of a complete binary tree with 2000 nodes

Solution: Using the formula from Example 6: $$h = \lfloor \log_2(2000) \rfloor$$

Calculate: $\log_2(2000) \approx 10.97$

Therefore: $h = \lfloor 10.97 \rfloor = 10$

Answer: Height = 10

Arithmetico-geometric Series

Formula: $\sum_{i=1}^{n} ir^i < \frac{r}{(1-r)^2}$ for $n \geq 1$, $0 < r < 1$

Example 8: Find an upper bound for the sum $\sum_{i=1}^{\infty} i(\frac{1}{2})^i$

Solution: This is an arithmetico-geometric series with r = 1/2.

Using the formula for the infinite sum: $$\sum_{i=1}^{\infty} ir^i = \frac{r}{(1-r)^2}$$

With r = 1/2: $$\sum_{i=1}^{\infty} i(\frac{1}{2})^i = \frac{\frac{1}{2}}{(1-\frac{1}{2})^2} = \frac{\frac{1}{2}}{(\frac{1}{2})^2} = \frac{\frac{1}{2}}{\frac{1}{4}} = \frac{1}{2} \times \frac{4}{1} = 2$$

Answer: Upper bound = 2

Course content developed by Declan Gray-Mullen for WNEU with Claude