CHAPTER 2

Section 2.5

Summation Notation

- Appendix B talks about sums
- Σ stands for summation, or sum
- General notation for a sum is

$$\sum_{i=p}^{q} (expression)$$

- Example

$$\sum_{i=1}^{n} (2i - 1) = 1 + 3 + 5 + \cdots + (2n - 1)$$

More Sums

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

$$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$$

$$\sum_{i=0}^{k} \frac{1}{2^i} = 2 - \frac{1}{2^k}$$

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$$

$$\sum_{i=0}^{k} i2^i = (k-1)2^{k+1} + 2$$

$$\sum_{i=0}^{k} a^i = \frac{a^{k+1} - 1}{a - 1}$$

Rules of Summation

$$\sum_{i=p}^{q}(a_i + b_i) = \sum_{i=p}^{q} a_i + \sum_{i=p}^{1} b_i$$

$$\sum_{i=p}^{q}(a_i - b_i) = \sum_{i=p}^{q} a_i - \sum_{i=p}^{1} b_i$$

$$\sum_{i=p}^{q} c a_i = c \sum_{i=p}^{q} a_i$$, where c is a constant

§ 2.5: Recurrence Relations

- Recall the recursive definition for S(n)
  1. $S(1) = 2$
  2. $S(n) = 2S(n\text{-}1)$ for $n \geq 2$
- We developed several algorithms for computing $S(n)$
- However, looking at the values in the sequence
  - 2, 4, 8, 16, 32, …
  - $2^1, 2^2, 2^3, 2^4, 2^5, \ldots$
  - We can see that $S(n) = 2^n$, which is a closed-form solution to the recurrence relation in (2) subject to the basis step.

Recurrence Relations

- Finding a closed-form solution to a recurrence relation is **solving** the recurrence relation.
  - Whenever possible we want to find a closed-form solution to a recurrence relation.
  - This section describes how to do so.

Expand, Guess, and Verify

- A technique for solving linear first-order recurrence relations.
  - Repeatedly use the recurrence relation to **expand** the expression for the nth term
  - **Guess** the general pattern
  - **Verify** the guess by mathematical induction

Example 42

- Same recurrence relation

  $S(1) = 2$

  $S(n) = 2S(n\text{-}1)$ for $n \geq 1$

- Expand

$$S(n) = 2S(n\text{-}1)$$
$$= 2[2S(n\text{-}2)] = 2^2S(n\text{-}2)$$
$$= 2[2[2S(n\text{-}3)]] = 2^3S(n\text{-}3)$$

Example 42 continued

- Guess the closed-form based on the developing pattern

  $S(n) = 2^k S(n-k)$

  - Must stop when $S(n-k) = S(1) \Rightarrow n-k=1 \Rightarrow k=n-1$

- At that point ($k=n-1$) the formula is

  $$S(n) = 2^{n-1}S(n-(n-1)) = 2^{n-1}S(n-n+1) = 2^{n-1}S(1)$$
  $$= (2^{n-1})2 = (2^{n-1+1}) = 2^n$$

  - This is our closed form solution

Example 42

- Prove using mathematical induction

Practice 21

- Find a closed-form solution for the recurrence relation, subject to the basis step, for sequence T

  $T(1) = 1$

  $T(n) = T(n-1) + 3$ for $n \geq 2$

Solution Formula

- General linear recurrence relation has the form

  $S(n) = f_1(n)S(n\text{-}1) + f_2(n)S(n\text{-}2) + \ldots + f_k(n)S(n\text{-}k) + g(n)$

  - Where the f_i' s and g' s can be expression involving n.
- Recurrence relation has **constant coefficients** if the f_i' s are all constants.
- It is **first-order** if the nth term depends only on term n-1
- Therefore, linear first-order recurrence relations with constant coefficients have the form

  $S(n) = cS(n\text{-}1) + g(n)$

Solution Formula

- Linear first-order recurrence relations with constant coefficients have the form

  $S(n) = cS(n-1) + g(n)$ (1)

- Which has the general solution

$$S(n) = c^{n-1}S(1) + \sum_{i=2}^{n} c^{n-i} g(i)$$

- To solve a recurrence relation of this form
  - Match the recurrence relation to formula (1)
  - Determine your values for c and $g(n)$
  - Substitute those values into the solution formula and solve

Practice 21

- Find a closed-form solution for the recurrence relation

  $T(1) = 1$

  $T(n) = T(n\text{-}1) + 3$ for $n \geq 2$
- Solution Formula method
- Alternate method
  - Set up the problem as a system of equations and add to solve

Example 42

- Recurrence relation

  $S(1) = 2$

  $S(n) = 2S(n\text{-}1)$ for $n \geq 1$

- Solution Formula method
- Alternate method

More Recurrence Relations

- Example 47: Solve the following recurrence relation

  $T(1) = 1$

  $T(n) = 2nT(n\text{-}1) + (n+1), \qquad n \geq 2$

Linear 2nd Order Recurrences

- The general form is $S(n) = c_1S(n\text{-}1) + c_2S(n\text{-}2)$
- The solution formula is

$$S(n) = pr_1^{n-1} + qr_2^{n-1}$$

- where r_1 and r_2 are two distinct roots of

$$t^2 - c_1t - c_2 = 0$$

- p and q must satisfy the two initial conditions

  $p + q = S(1)$

  pr_1 to $qr_2 = S(2)$

Log Function (Appendix C)

- $y = \log_b x$ means $b^y = x$
- Properties of the Logarithm Function $y = \log_b x$

1. If $p < q$, then $\log_b p < \log_b q$
2. If $\log_b p = \log_b q$, then $p = q$
3. $\log_b 1 = 0$
4. $\log_b b = 1$
5. $\log_b(b^p) = \mathrm{p}$
6. $\log_b(pq) = \log_b p + \log_b q$
7. $\log_b(p/q) = \log_b p - \log_b q$
8. $\log_b(p^q) = q(\log_b p)$
9. $\log_a \mathrm{p} = \log_b p / \log_b a$

Divide and Conquer Recurrences

- Divide and conquer recurrence relations with constant coefficients have the form

$$S(n) = cS(n/2) + g(n) \qquad , n \geq 2,\ \mathrm{n}{=}2^m \qquad (1)$$

- Which has the general solution

$$S(n) = c^{\log n} S(1) + \sum_{i=1}^{\log n} c^{(\log n)-i} g(2^i)$$

Divide and Conquer Recurrences

- Example 52: Solve the following recurrence relation

$$C(1) = 1$$

$$C(n) = 1 + C(n/2), \qquad n \geq 2, n=2^m$$

Divide and Conquer Recurrences

- Problem 40: Solve the following recurrence relation

$$S(1) = 1$$

$$S(n) = 2S(n/2) + n, \qquad n \geq 2,\ n=2^m$$