Finding close forms

Lecture 20
ICOM 4075
What is a close form?

In the previous lecture we discuss the relation between the size of the input and the complexity of computing the output. Close forms are instances in which such relation is constant.

**Definition:** A close form is an expression that can be computed by applying a fixed number of operations to the inputs.
Example

**Problem**: Find the sum of the first n naturals, where n is a given natural.

**Closed form solution**:

\[ \text{Close}(n) \]

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

Return \( s \)

* Takes 3 arithmetic operations: complexity is 3, it doesn’t grow!*

**Non – closed form solutions**: May be inductive or recursive.

**Inductive solution**:

\[ \text{Inductive}(n) \]

\[ s \leftarrow 0 \]

For i = 1 to n

\[ s \leftarrow s + i \]

Return \( s \)
Non – close solutions

Recursive solution:

\[ s \leftarrow n \]

Recursive\((s, n)\)

- If \(n = 1\)
  - Return \(s\)
- Else \(n \leftarrow n - 1\)
  - \(s \leftarrow s + n\)
  - Recursive\((s, n)\)

Complexity:

Both, inductive and recursive algorithms perform \(n\) additions

So, they both grow \textit{linearly} with \(n\)
Close forms of elementary sums

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

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

c. \[ \sum_{i=0}^{n} a^i = \frac{a^{n+1}-1}{a-1}; \quad a \neq 1 \]

d. \[ \sum_{i=1}^{n} ia^i = \frac{a-(n+1)a^{n+1}+na^{n+2}}{(a-1)^2}; \quad a \neq 1 \]
Their recursive counterparts

a. \( s_1 = 1 \)
   \[ s_n = s_{n-1} + n, \ n > 1 \]

b. \( s_1 = 1 \)
   \[ s_n = s_{n-1} + n^2, \ n > 1 \]

c. \( s_0 = 1 \)
   \[ s_n = s_{n-1} + a^n, \ n > 1, \ a \neq 1 \]

d. \( s_1 = a \)
   \[ s_n = s_{n-1} + na^n, \ n > 1, \ a \neq 1 \]

All the corresponding algorithms grow linearly with \( n \), as well
Example of direct computations

\[ \sum_{i=1}^{35} i^2 = \frac{35 \cdot (35+1) \cdot (2 \cdot 35+1)}{2} = \frac{89460}{2} = 44730 \]

\[ \sum_{i=1}^{9} i \cdot 3^i = \frac{3 - (9+1)3^{9+1} + 9 \cdot 3^{9+2}}{(3-1)^2} = \frac{3 - 10 \cdot 59049 + 9 \cdot 177147}{4} = \frac{1003836}{4} = 250959 \]
Finding new close forms

Some new close forms can be derived by combining the elementary sums with the following operation rules

1. \[ \sum_{i=m}^{n} c = (n - m + 1)c \]

2. \[ \sum_{i=m}^{n} a_i + b_i = \sum_{i=m}^{n} a_i + \sum_{i=m}^{n} b_i \]

3. \[ \sum_{i=m}^{n} ca_i = c \sum_{i=m}^{n} a_i \]

4. \[ \sum_{i=m}^{n} a_{i+k} = \sum_{i=m+k}^{n+k} a_i, \text{ } k \text{ is a fixed integer} \]
Example

Problem: For any given natural number \( n \), find the sum of \( 30i^2 - 23i + 427 \), for \( i = 1, \ldots, n \).

A close form algorithmic solution is:

\[
\sum_{i=1}^{n} 30i^2 - 23i + 427 = 30 \sum_{i=1}^{n} i^2 - 23 \sum_{i=1}^{n} i + \sum_{i=1}^{n} 427 \text{ (rules 2 and 3)}
\]

\[
= 30 \frac{n(n+1)(2n+1)}{2} - 23 \frac{n(n+1)}{2} + 427n
\]

\[
= \frac{n(n+1)}{2} (30 \cdot (2n+1) - 23) + 427n
\]

\[
= \frac{n(n+1)}{2} (60n + 7) + 427n
\]
More examples

**Problem:** Find the sum of the first $n$ odd numbers

**Answer:** Using the rules and basic sums we get the close formula

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

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

$$
= n^2 + 2n
$$

With this formula, any such sum can be computed with three operations (two multiplications and one addition)
Adding higher powers

Finding a close form for an inductive or recursive summation (if there such close form) is usually tricky. For example, here is a derivation of a close form for \(\sum_{i=1}^{n} i^3\).

\[
\sum_{i=1}^{n} i^4 + (n + 1)^4 = \sum_{i=0}^{n} (i + 1)^4
\]

\[
= \sum_{i=0}^{n} \left( i^4 + 4i^3 + 6i^2 + 4i + 1 \right)
\]

\[
= \sum_{i=1}^{n} i^4 + 4\sum_{i=1}^{n} i^3 + 6\sum_{i=1}^{n} i^2 + 4\sum_{i=1}^{n} i + \sum_{i=1}^{n} 1
\]

Now, by cancelling out the summations of the fourth powers of \(i\) one gets (next slide)
Continuation

\[(n+1)^4 = 4 \sum_{i=1}^{n} i^3 + 6 \sum_{i=1}^{n} i^2 + 4 \sum_{i=1}^{n} i + \sum_{i=0}^{n} 1\]

\[= 4 \sum_{i=1}^{n} i^3 + 6 \frac{n(n+1)(2n+1)}{2} + 4 \frac{n(n+1)}{2} + (n+1)\]

\[= 4 \sum_{i=1}^{n} i^3 + (n+1)(3n(2n+1) + 2n+1)\]

\[= 4 \sum_{i=1}^{n} i^3 + (n+1)(5n^2 + 5n + 1)\]

Therefore,

\[\sum_{i=1}^{n} i^3 = \frac{(n + 1)((n + 1)^3 - 20(n^2 + n) - 4)}{4}\]

\[= \frac{(n + 1)((n + 1)[(n + 1)^2 - 20n] - 4)}{4}\]
Method

The previous derivation is indeed an application of the following general method for finding a close form for the sum of $n$ terms of the form $i$ the $k$th power. The method assumes that such close forms are known for all powers $q < k$.

1. Write the equation $\sum_{i=1}^{n} i^{k+1} + (n + 1)^{k+1} = \sum_{i=0}^{n} (i + 1)^{k+1}$

2. Develop the general term of the right hand side summation, express it as a polynomial

3. Cancel out $\sum_{i=1}^{n} i^{k+1}$ from each side of the equation

4. Replace all close forms formulas for powers $q < k$

5. Solve the equation for $\sum_{i=1}^{n} i^{k}$
Use of close formulas in algorithm analysis

Let $a[i]$ be an array of natural numbers and $\text{min}(a, i, n)$ the index of the minimum number $a[i], a[i+1], \ldots, a[n]$

Consider the following sorting algorithm:

```
Simple(a, n)
For i = 1 to n - 1
  j ← $\text{min}(a, i, n)$
  swap(a[i], a[j])
Return a
```
What is the rate of growth of Simple?

Complexity count:

```
Simple(a, n)
For i = 1 to n – 1
  j ← min(a, i, n)
  swap(a[i], a[j])
Return a
```

The operation of \textit{min} involves \(n - i\) comparisons. Since \(i\) ranges from 1 to \(n - 1\) we get \(\sum_{i=1}^{n-1} n - i\). What type of growth is this? (linear, polynomial, exponential,...)
The rate of growth of Simple

Using the previous techniques, we transform the summation into a closed form, to get

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

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

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

So, the method grows quadratically
Exercises

1. Find a close form for \( \sum_{i=0}^{n} 25i^3 \frac{(3i+7)^3}{32} \)

2. Find a close form for the following method:
   Input: n (a natural number)
   A \leftarrow 0
   For j = 0 to n
     B \leftarrow j \cdot j
     A \leftarrow A + 2 \cdot B \cdot B + B
   Return A
   Compare the complexity of the method with that of calculating the close form