Today: 09/16-ribbon

- Asymptotic Complexity

Reminder:
- A2 due today

Heads-Up:
- As out soon
- Prelim in 16/15 days

Counting steps in a program

How many instructions are executed.

Ex: for (int i = 0; i < n; i++) {
    for (int j = 0; j < i; j++)
        print 'x';
        printIn;
}

1: assignment, n+1 comp, n increments
2: 1 assignment, int comp, i increments
3: 1 op
4: 1 op
5: Inner for loop: increment takes two steps!
   1 + (n+1) + 2i + 1 = Ai + 2 steps

Outer for loop:
   1 + (n+1) + 2n + \sum_{i=0}^{n-1} (ai+2i)
   = 3(n+2) + 3n + \sum_{i=0}^{n-1} i
   = 6n + 2 + 2n(n-1)
   = 2n^2 + 4n + 2 steps

Remove println?
   \to a bit faster

Remove inner for loop?
   \to a lot faster

\* Care about dominating parts of the program.
   \Add print in inner for loop?
   \to proportionally slower

\* Constant steps don't matter

Asymptotic Complexity

Describes dominating factor in a function.

Big-O notation: Bounds the function from above.

Ex: 2n^2 + 4n + 2 is O(n^2), also O(c * n^2).

Def: \( f(n) \) is \( O(g(n)) \) if there exist
   - positive constant \( k \)
   - natural number \( n_0 \)
   such that \( f(n) \leq kg(n) \) for all \( n > n_0 \).

Ex: \( 2n^2 + 4n + 2 \leq 2n^2 + 4n^2 + 2n^2 = 8n^2 \) for \( n \geq 1 \).
   \( k=2, n_0=1 \) \( \square \)

\( 2n^2 + 4n + 2 \leq 8n^2 \) for \( n \geq 1 \).
   \( k=8, n_0=1 \) \( \square \)

Shortcut: \( f(n) \) is \( O(g(n)) \) if \( \lim_{n \to \infty} \frac{f(n)}{g(n)} \) exists and \( < \infty \).

Ex:
   \( \lim_{n \to \infty} \frac{2n^2 + 4n + 2}{n^2} = 2 \) \( \lim_{n \to \infty} \frac{9n^2 + 4n + 2}{n^3} = 0 \)

Ex: Is \( n^2 \) \( O(n) \)?

No - show that no witness exists.

\=> Given a witness, fail it.

\="Prove by contradiction."

Suppose \( n^2 \leq kn \) for all \( n \geq n_0 \).

Choose \( n_x = \max(k, n_0) + 1 \). So, \( n_x > k \).

Then, \( n_x > kn_x \).

Thm: If \( f_1(n) \) is \( O(g_1(n)) \) and \( f_2(n) \) is \( O(g_2(n)) \), then
   \( f_1(n) + f_2(n) \) is \( O(g_1(n) + g_2(n)) \).

Also:

Ex: loop:
   \[ \frac{1}{c_{f_0}} \left[ O(\sum_{i=0}^{n-1} i) \right] ^{n} \in \left[ O(1) \right] ^{\log n} \times \left[ O(n^2) \right] \]

Families of Functions:

- \( O(1) \) constant
- \( O(n) \) linear
- \( O(n^2) \) quadratic
- \( O(n^3) \) polynomial
- \( O(k^n) \) exponential
Ex. 
getConsecutiveSum(int n)

for (int i=1; i<=n; i++) {
    int sum=0;
    for (int j=i; sum<n; j++)
        sum += j;
    if (sum==n)
        // report
}

Inner loop: $O\left(\frac{n^2}{2}\right)$
Outer loop: $O\left(\sum_{i=1}^{n} \frac{n}{i}\right)$

= $O\left(n \sum_{i=1}^{n} \frac{1}{i}\right)$

= $O\left(n \log n\right)$

2. Want $x$ such that for some $k$
\[ x + (x+1) + \cdots + (x+k-1) = n \]
\[ kx + \sum_{i=1}^{k-1} i = n \]
\[ kx + \frac{k(k-1)}{2} = n \]
\[ \Rightarrow \left(n - \frac{k(k-1)}{2}\right) \mod k = 0. \]

int sumk=0;
for (int k=1; sumk<n; k++) {
    if (\((n-sumk) \mod k == 0\)) {
        int x = (n-sumk)/k;
        // report
    }
    sumk += k;
}

Loop: $k_{\text{max}}$ gives $\text{sumk} = \frac{k_{\text{max}}(k_{\text{max}}-1)}{2} > n$

\[ k_{\text{max}} \approx 2.41 \sqrt{n} \]

$\Rightarrow O(\sqrt{n})$