Today: 09/16-17/2014

- Asymptotic Complexity

Reminder:

- A2 due today

Heads-Up:

- A3 out soon
- Prelim in 16/15 days

Counting steps in a Program

How many instructions are executed.

Ex,
```
1  for (int i=0; i<n; i++) {
2      for (int j=0; j<i; j++)
3          print '*';
4      println;
5  }
```

1 1 assignment, n+1 comp, n increments

2 1 assignment, i+1 comp, i increments

3 1 op

4 1 op

5

Inner for loop: increment takes two steps!

$$1 + (i+1) + 2i + \sum_{j=0}^{i-1} 1 = 4i + 2 \quad \text{steps}$$

Outer for loop:

$$1 + (n+1) + 2n + \sum_{i=0}^{n-1} (4i+2+1)$$

$$= 3n+2 + 3n + 4\sum_{i=0}^{n-1} i$$

$$= 6n+2 + 2n(n-1)$$

$$= 2n^2+4n+2 \quad \text{steps}$$

Remove println?

→ a bit faster

Remove inner for loop?

→ a lot faster

* Care about dominating parts of the program.

Add print in inner for loop?

→ 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 $O(n^3)$.

Def: $f(n)$ is $O(g(n))$ if there exist

- positive constant k
- natural number n_0

} witness

such that $f(n) \leq k g(n)$ for all $n \geq n_0$.

Ex, $2n^2+4n+2 \leq 2n^2+4n^2+2n^2 = 8n^2$ for $n \geq 1$.

$k=8,\ n_0=1$ $\boxed{\text{is } O(n^2)}$

$2n^2+4n+2 \leq 8n^3$ for $n \geq 1$

$k=8,\ n_0=1$ $\boxed{\text{is } O(n^3)}$

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

Ex, $\lim_{n\to\infty} \frac{2n^2+4n+2}{n^2} = 2$, $\lim_{n\to\infty} \frac{2n^2+4n+2}{n^3} = 0$

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, $n_x^2 > kn_x$ ↯

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 $O(g_1(n) + g_2(n))$ $\boxed{\text{Also} \cdot}$

Ex, loop 1 2 3 4 5

$$O\left(\sum_{i=0}^{n} i\right) = O(n^2) \quad \left[\; O(i) \left[\, O(1) \,\right\} i \text{ times} \quad O(1) \;\right\} n \text{ times}$$

Families of Functions

$O(1)$	constant
$O(n)$	linear
$O(n^2)$	quadratic
$O(n^k)$	polynomial
$O(k^n)$	exponential

Ex. getConsecutiveSum(int n)

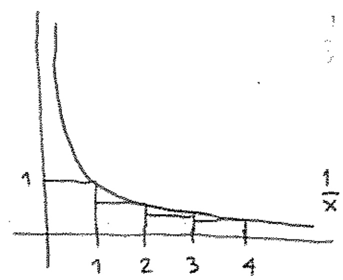
①
```
for (int i=1; i<=n; i++) {
    int sum=0;
    for (int j=i; sum<n; j++)
        sum += j;
    if (sum==n)
        // report
}
```

Inner loop: $O(\frac{n}{i})$

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

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

$= O(n \log n)$



$\sum_{i=1}^{n} \frac{1}{i} < 1 + \int_1^n \frac{1}{x} dx$

$= 1 + \ln n$

$= 1 + \frac{\log n}{\log e}$

is $O(\log n)$

$n = 10000$

$\rightarrow 14 \times 10^8$

② Want x such that for some k

$x + (x+1) + \cdots + (x+k-1) = n$

$kx + \sum_{i=1}^{k-1} i = n$

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

$\Rightarrow \left(n - \frac{k(k-1)}{2}\right) \bmod k = 0.$

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

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

$k_{max} \approx 2\sqrt{n}$

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

$n = 10000$

$\rightarrow 100$