There are a couple of goals for this section.

- To remind you of the basics of expressing the run time of an iterative algorithm using a summation.
- To remind you of some of the most commonly-used identities for simplifying summations.
- To demonstrate a few ‘tricks’ that can be used to solve many summations that occur in the analysis of algorithms.

1 Definition

A common tool for analyzing iterative algorithms is the summation:

$$\sum_{i=\ell}^{u} a_i = a_\ell + a_{\ell+1} + \cdots + a_{u-1} + a_u$$

If the upper limit is infinite, we interpret this as an implicit limit:

$$\sum_{i=\ell}^{\infty} a_i = \lim_{u \to \infty} \sum_{i=\ell}^{u} a_i$$

2 Two examples

Do these two algorithms have the same asymptotic running time?

```c
int AlgA(int n) {
    int sum = 0;
    for(int i=0; i<n; i++) {
        for(int j=0; j<i; j++) {
            sum++;
        }
    }
    return sum;
}
```

```c
int AlgB(int n) {
    int sum = 0;
    for(int i=0; i<n; i++) {
        for(int j=0; j<n; j++) {
            sum++;
        }
    }
    return sum;
}
```

3 Same asymptotic run times

Yes, the run times are both $\Theta(n^2)$.
\[ A(n) = \sum_{i=0}^{n-1} \sum_{j=0}^{i-1} 1 = \sum_{i=0}^{n-1} i = \frac{(n-1)n}{2} = \Theta(n^2) \]

\[ B(n) = \sum_{i=0}^{n-1} \sum_{j=0}^{i-1} 1 = \sum_{i=0}^{n-1} n = n^2 = \Theta(n^2) \]

### 4 Two more examples

Do these two algorithms have the same asymptotic running time?

```c
int AlgC(int n) {
    int sum = 0;
    for(int i=1; i<n; i*=2) {
        for(int j=0; j<i; j++) {
            sum++;
        }
    }
    return sum;
}
```

```c
int AlgD(int n) {
    int sum = 0;
    for(int i=1; i<n; i*=2) {
        for(int j=0; j<n; j++) {
            sum++;
        }
    }
    return sum;
}
```

### 5 Different asymptotic run times

No, the asymptotic run times are different. Observe that \( i \) is always a power of 2. Let \( i = 2^k \), so that \( k = \lg i \).

\[ C(n) = \sum_{k=0}^{\lfloor \lg n \rfloor} \sum_{j=0}^{2^k-1} 1 = \sum_{k=0}^{\lfloor \lg n \rfloor} 2^k \]
\[ = \sum_{k=0}^{\lfloor \lg n \rfloor} 2^k = 2^{\lfloor \lg n \rfloor} - 1 = \Theta(n) \]

\[ D(n) = \sum_{k=0}^{\lfloor \lg n \rfloor} \sum_{j=0}^{n-1} 1 = \sum_{k=0}^{\lfloor \lg n \rfloor} n \]
\[ = \lfloor \lg n \rfloor n = \Theta(n \log n) \]

### 6 Some common summations

The arithmetic series:
\[ \sum_{i=1}^{n} i = \frac{n(n+1)}{2} = \Theta(n^2) \]

Sums of squares, cubes, and higher powers:
\[ \sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} = \Theta(n^3) \]
$$\sum_{i=1}^{n} i^3 = \frac{n^2(n+1)^2}{4} = \Theta(n^4)$$

$$\sum_{i=1}^{n} i^k = \Theta(n^{k+1})$$

7 Proof for $\sum i^k$

To show that $\sum_{i=1}^{n} i^k = \Theta(n^{k+1})$, we need both upper and lower bounds.

Upper bound:
$$\sum_{i=1}^{n} i^k \leq \sum_{i=1}^{n} n^k = n \cdot n^k = n^{k+1}$$

Lower bound:
$$\sum_{i=1}^{n} i^k \geq \sum_{i=[n/2]}^{n} i^k \geq \sum_{i=[n/2]}^{n} \left(\frac{n}{2}\right)^k = \frac{n}{2} \left(\frac{n}{2}\right)^k = \frac{1}{2^{k+1}}n^{k+1}$$

Therefore, we can use $c_1 = \frac{1}{2^{k+1}}$ and $c_2 = 1$ as the constants in the $\Theta$ definition.

8 Finite geometric series

For constants $r$ and $n$, the finite geometric series is:
$$\sum_{i=0}^{n-1} r^i = \frac{r^n - 1}{r - 1}$$

9 Deriving the finite geometric series formula

To prove this closed form is correct, let $S$ denote the sum. We have:
$$S - rS = \sum_{i=0}^{n-1} r^i - r \sum_{i=0}^{n-1} r^i$$
$$= \sum_{i=0}^{n-1} r^i - \sum_{i=0}^{n-1} r^{i+1}$$
$$= \sum_{i=0}^{n-1} r^i - \sum_{i=1}^{n} r^i$$
$$= 1 - r^n$$
Then solve for $S$:

$$S = \frac{r^n - 1}{r - 1}$$

10 Infinite geometric series

For a constant $r < 1$, the infinite geometric series is:

$$\sum_{i=0}^{\infty} r^i = \frac{1}{1 - r}$$

11 Deriving the infinite geometric series formula

We can use the finite version, along with the definition of an infinite summation, to evaluate this:

$$\sum_{i=0}^{\infty} r^i = \lim_{n \to \infty} \sum_{i=0}^{n-1} r^i = \lim_{n \to \infty} \frac{1 - r^n}{1 - r} = \frac{1}{1 - r} - \frac{1}{1 - r} \lim_{n \to \infty} r^n = \frac{1}{1 - r}$$

12 Bounding with integrals

Sometimes, an integral may be easier to evaluate than a sum.

If $f(n)$ is monotone increasing:

$$\int_{m-1}^{n} f(x) \, dx \leq \sum_{i=m}^{n} f(n) \leq \int_{m}^{n+1} f(x) \, dx$$
If $f(n)$ is monotone decreasing:

$$\int_{m}^{n+1} f(x) \, dx \leq \sum_{i=m}^{n} f(n) \leq \int_{m-1}^{n} f(x) \, dx$$