Summations

- Most nontrivial algorithms have iterative control structures.
- So dominating term for running time is determined loops.
- Thus the running time of an algorith in most cases expressed as a sum of running time of each execution of loop body.
- Thus finding bounds for various sums is quite important for analysis of running time.

Example

```
int maxSubSeqSum(int a[], int n) {
    int current=0;
    int maxSum=0;
    for (int i=0; i<n; i++)
        for (int j=i; j<n; j++) {
            current = 0;
            for (int k=i; k<= j; k++)
                current += a[k]
            if (current > maxSum)
                maxSum = current;
        }
    return maxSum;
}
```


Example

- Loop at L4 is executed n times.
- Second loop at L5 is executed $n - i + 1$ times for each i.
- Third loop at L7 is executed $j - i + 1$ times for each j and each i.
- But L5 and L7 in worst case can be executed n times each.
- So, worst case time of running is $O(n^3)$.

More Precise Analysis

- Actual sum of running time

$$\sum_{i=1}^{n} \sum_{j=i}^{n} \sum_{k=i}^{j} 1$$

- We know

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

- And

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

- Thus, summing RHS expression

$$\tfrac{1}{2}\sum_{i=1}^{n} i^2 - \left(n - \tfrac{3}{2}\right)\sum_{i=1}^{n} i + \tfrac{1}{2}(n^2+3n+2)\sum_{i=1}^{n} 1$$
$$= \frac{1}{2}\frac{n(n+1+(2n+1)}{6} - \left(n - \frac{3}{2}\right)\frac{n(n+1)}{2} + \frac{n^2+3n+2}{2}n$$
$$= \frac{n^3+3n^2+2n}{6}$$

Improving Example

- Notice in earlier example sum $\sum_{k=i}^{j} a[k]$ recomputed every time from scratch when j is incremented.
- But since $\sum_{k=i}^{j} a[k] = a[j] + \sum_{k=i}^{j-1} a[k]$, the innermost loop can be eliminated.
- With the suggested changes, the complexity is $O(n^2)$.
- It can be improved further, but the discussion is restricted to the role of sums in analysis of running time.

Improving Example

```
int maxSubSeqSum(int a[], int n) {
    int current=0;
    int maxSum=0;
    for (int i=0; i<n; i++) {
        current = 0;
        for (int j=i; j<n; j++) {
            current += a[j]
            if (current > maxSum)
                maxSum = current;
        }
    }
    return maxSum;
}
```

Sums: Common Formulas

- Arithmetic series – sum of natural numbers.
- Geometric series – infinite decreasing GS.
- Harmonic series.

Sum Manipulations

- Differntiation.
  - Consider GS
  $$\sum_{k=0}^{\infty} x^k = \frac{1}{1-x} \text{ where } |x| < 1$$
  - Differentiation of both sides and multiplication by x gives
  $$\sum_{k=0}^{\infty} kx^k = \frac{x}{(1-x)^2}$$

Sum Manipulations

- Telescoping

$$\sum_{k=1}^{n} (a_k - a_{k-1}) = a_n - a_0$$

$$\sum_{k=0}^{n-1} (a_k - a_{k+1}) = a_0 - a_n$$

  - Consider

$$\begin{aligned}\sum_{k=1}^{n-1} \frac{1}{k(k+1)} &= \sum_{k=1}^{n-1} \left(\frac{1}{k} - \frac{1}{k+1}\right) \\ &= 1 - \frac{1}{n}\end{aligned}$$

Sum Manipulations

- Coverting product to sum
  - Consider

$$\log\left(\prod_{k=1}^{n} a_k\right) = \sum_{k=1}^{n} \log a_k$$

- Bounding sums
  - By mathematical induction:

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

  - Bounding terms

$$\sum_{k=1}^{n} k \le \sum_{k=1}^{n} n$$

Sum Manipulations

- Bounding sums
  - By bounding terms: in general

$$\sum_{k=1}^{n} a_k \leq n \max_k \{a_k\}.$$

Suppose $a_k \leq a_0 r^k$, then

$$\sum_{k=0}^{n} a_k \leq \sum_{k=0}^{n} a_0 r^k = a_0 \sum_{k=0}^{n} r^k = a_0 \frac{1}{1-r}$$

- Splitting summations

$$\sum_{k=1}^{n} k = \sum_{k=1}^{n/2} k + \sum_{k=(n/2)+1}^{n} k \geq \sum_{k=0}^{n/2} 0 + \sum_{k=(n/2)+1}^{n} (n/2)$$

Therefore,

$$\sum_{k=1}^{n} k \geq (n/2)^2 = \Omega(n^2)$$

Splitting Summation: Example

- Consider

$$\sum_{k=0}^{\infty} \frac{k^2}{2^k}$$

Since

$$\frac{(k+1)^2/2^{k+1}}{2^k/k} = \frac{(k+1)^2}{2k^2} \leq \frac{8}{9}, \text{ if } k \geq 3$$

We have

$$\sum_{k=0}^{\infty} \frac{k^2}{2^k} = \sum_{k=0}^{2} \frac{k^2}{2^k} + \sum_{k=3}^{\infty} \frac{k^2}{2^k}$$

$$\leq O(1) + \frac{8}{9} \sum_{k=0}^{\infty} \left(\frac{8}{9}\right)^k \leq O(1)$$

Harmonic Series

- Harmonic series $H_n = \sum_{k=1}^{n} \frac{1}{k}$
- A loose lower bound can be found as follows:

$$\begin{aligned} H_n &= 1 + \tfrac{1}{2} + \left(\tfrac{1}{3} + \tfrac{1}{4}\right) + \left(\tfrac{1}{5} + \tfrac{1}{6} + \tfrac{1}{7} + \tfrac{1}{8}\right) + \ldots \\ H_n &\geq 1 + \tfrac{1}{2} + 2 \times \tfrac{1}{4} + 4 \times \tfrac{1}{8} + \ldots \\ H_n &\geq 1 + \tfrac{\log n}{2} \end{aligned}$$

- Similarly

$$H_n \leq \sum_{i=0}^{\lceil \log n \rceil} \sum_{j=0}^{2^j - 1} \frac{1}{2^i + j} \leq \sum_{i=0}^{\lceil \log n \rceil} \sum_{j=0}^{2^j - 1} \frac{1}{2^i}$$

implies that

$$H_n \leq \sum_{i=0}^{\lceil \log n \rceil} 1 \leq \log n + 1$$

Approximation by Integrals

- When summations are of the form $\sum_{k=m}^{n} f(k)$ where $f(k)$ is a monotonically increasing function, intgrals can be used for approximation.

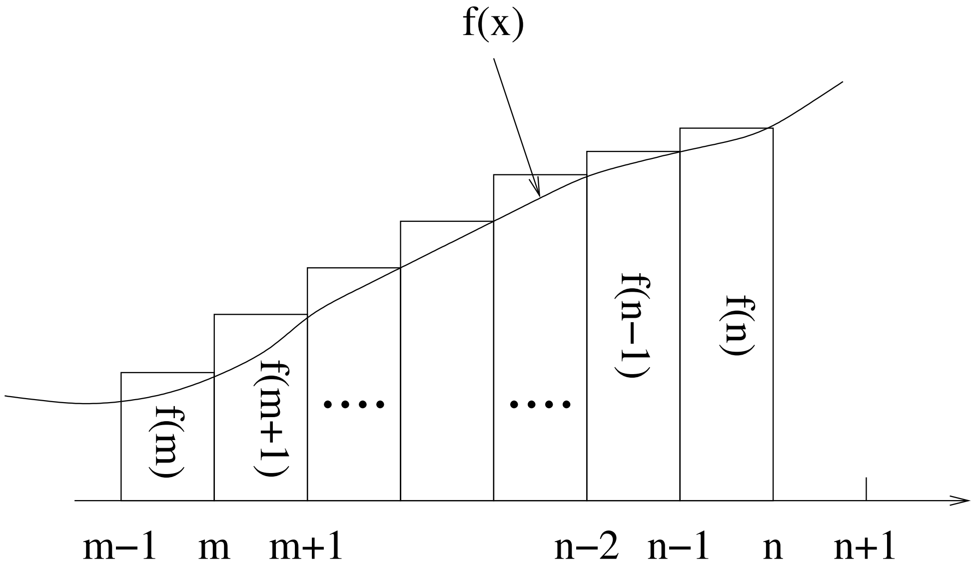
$$\int_{m-1}^{n} f(x)dx \leq \sum_{k=m}^{n} f(k) \leq \int_{m}^{n+1} f(x)dx$$

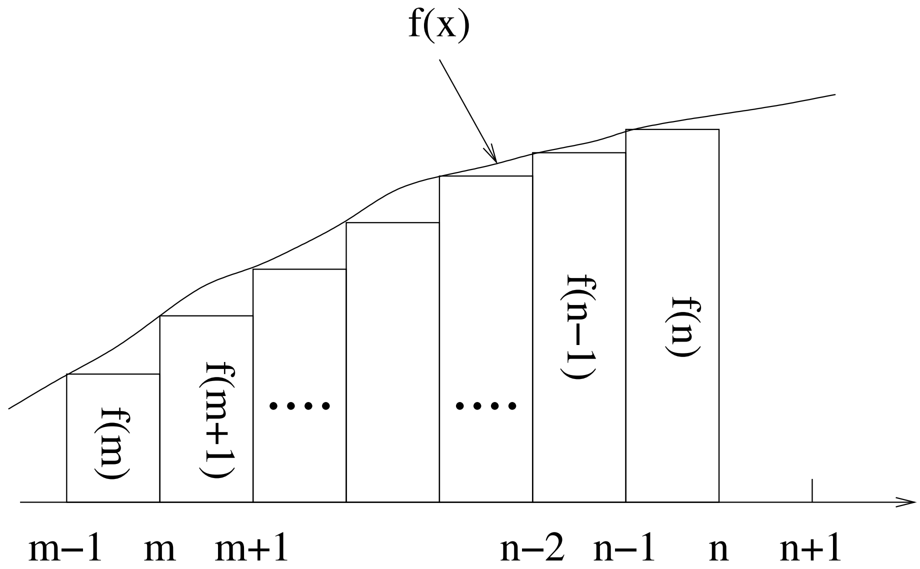
- If $f(k)$ is monotonically decreasing then

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

- So integrals can be used to find tighter bounds.
- See figure next for justification.

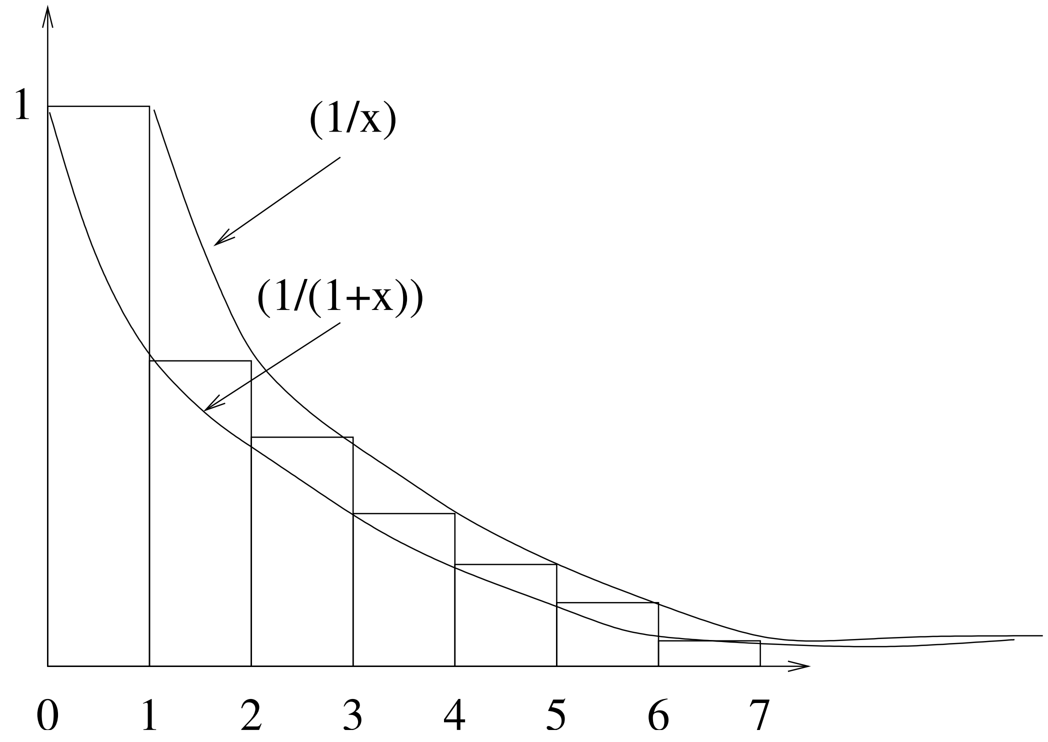
Approximation by Integrals





Summation

- Using integral approximation for nth harmonic sum we find



- Thus

$$\int_0^n \frac{1}{1+x}dx \le H_n \le 1 + \int_1^n \frac{1}{x}dx$$

Implying

$$\log n \le H_n \le 1 + \log n$$

Approximation by Integrals

- Consider $\sum_{i=1}^{n} i^2$
- Using integral approximation

$$\int_0^n x^2 dx \le \sum_{i=1}^{n} i^2 \le \int_1^{n+1} x^2 dx$$

$$\frac{n^3 - 1}{3} \le \sum_{i=1}^{n} i^2 \le \frac{n^3}{3} + n^2 + n$$

- We know

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} = \frac{n^3}{3} + \frac{1}{2}\, n^2 + \frac{1}{6}\, n$$

Sum Manipulations

- We have seen manipulation of sums important for analysis.
- The usual manipulation tricks are
  - Differentiations
  - Telescoping
  - Bounding terms
  - Splitting sums
  - Converting products to sums
  - Bounding by definite Integrals