Discrete Structures

Fight for Survival

Dr. Mudassir Shabbir

August 27, 2026

Solving Recurrences

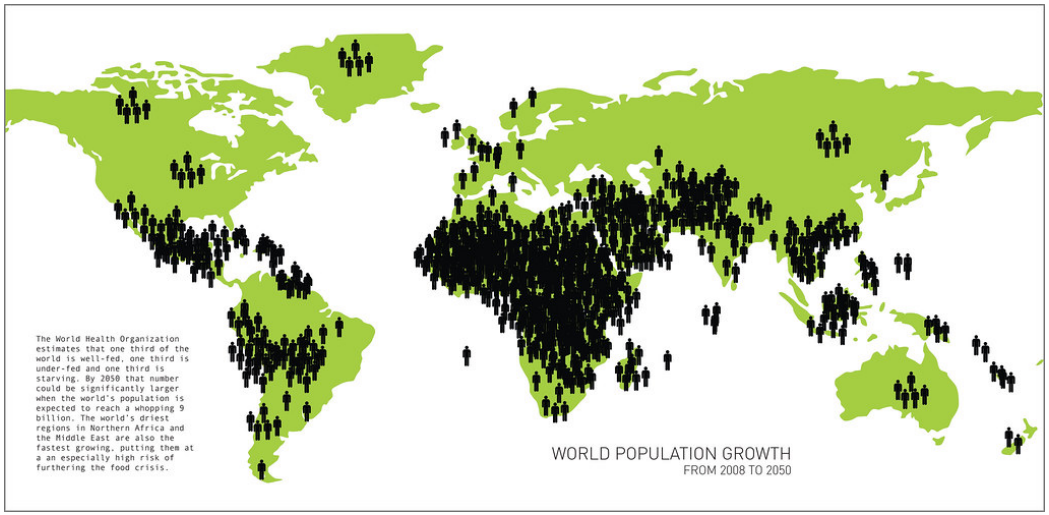
Recurrences

We observe that things are highly correlated to their immediate past!

Recurrences

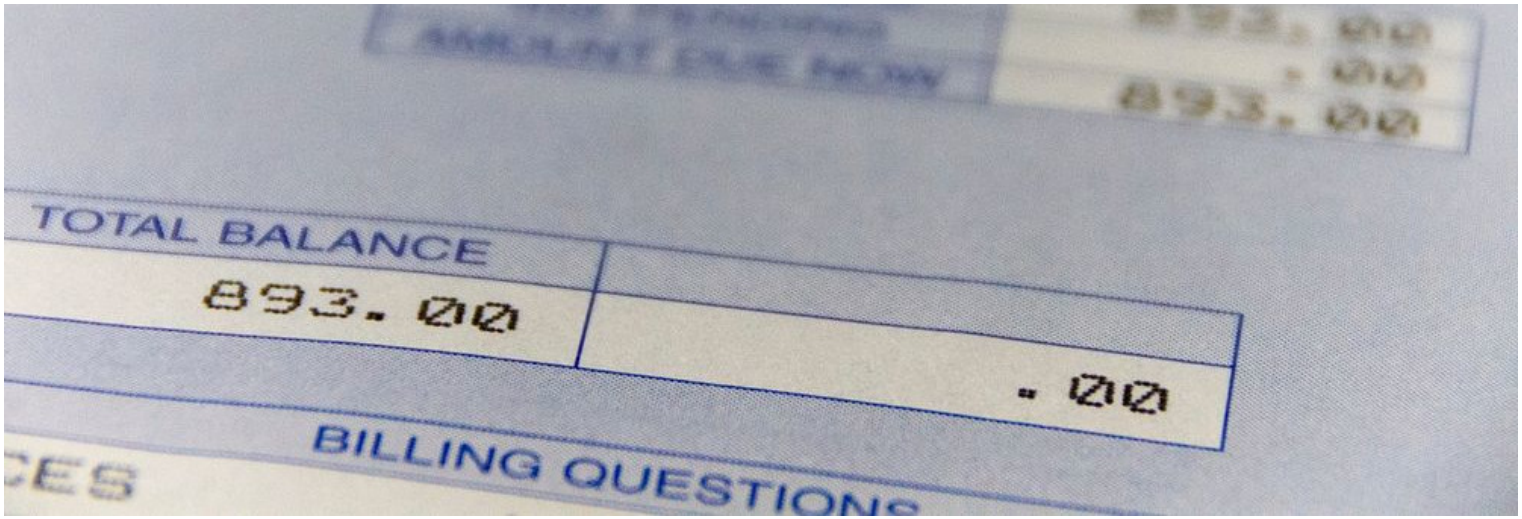
Temperature now is very close to what it was one minute ago!

Recurrences



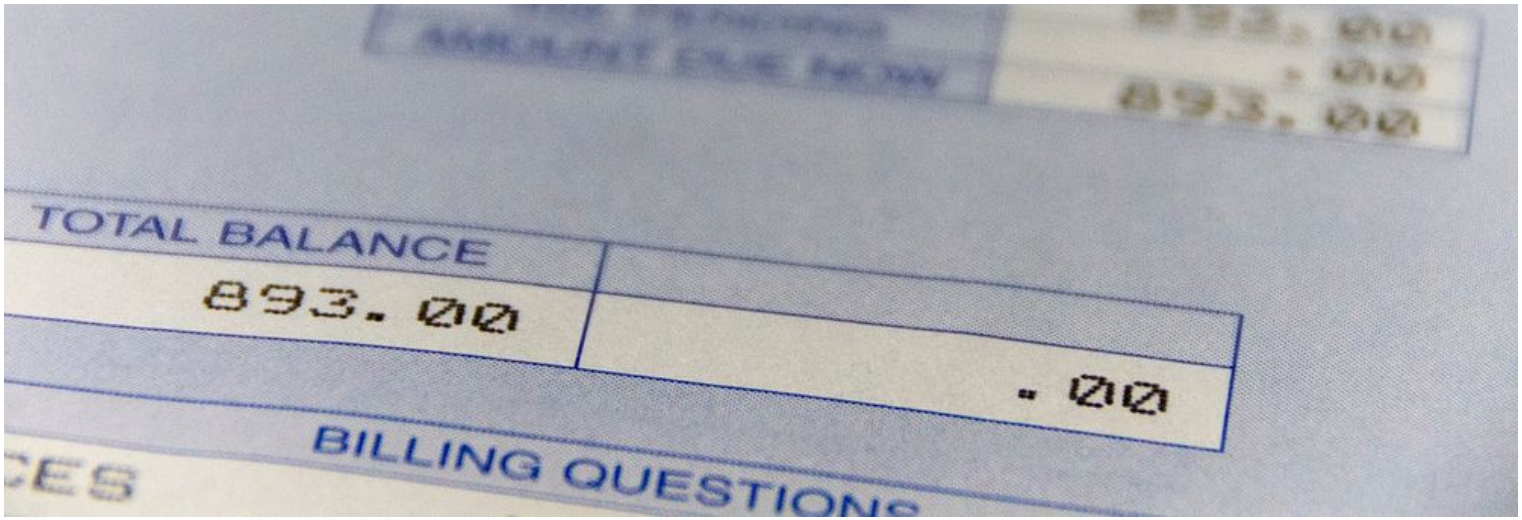
World Population is very reliable function of what it was a year ago!

Recurrences



Your bank account balance looks very much like what it was a month ago!

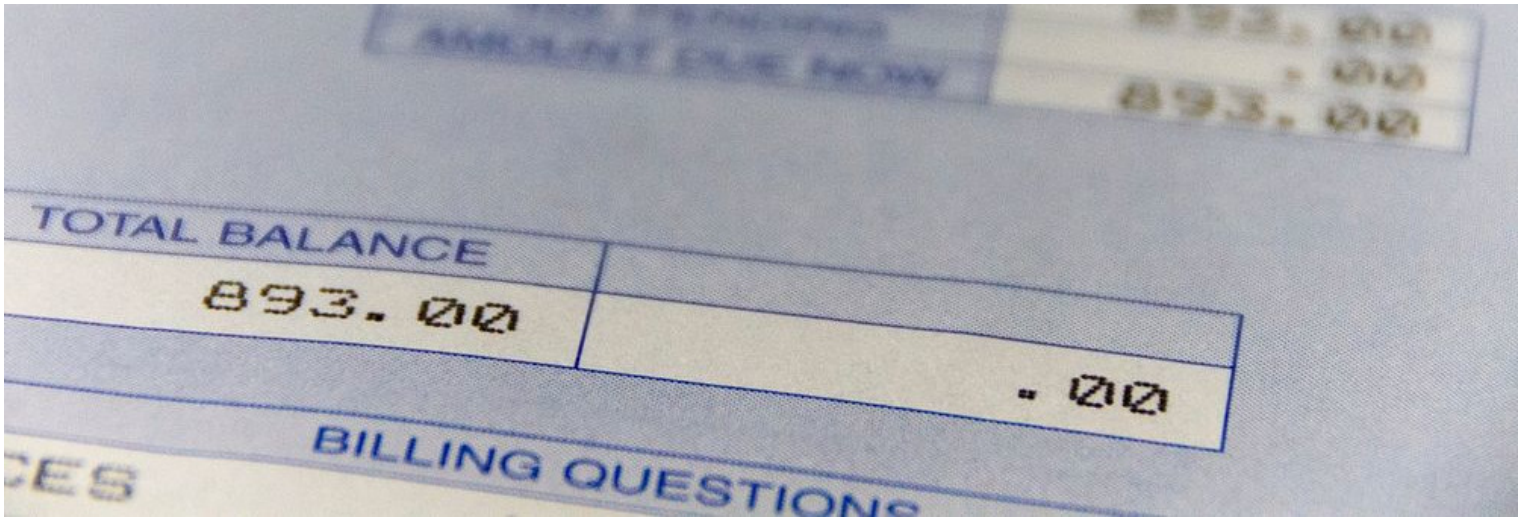
Recurrences



Your bank account balance looks very much like what it was a month ago!

We can reliably estimate current state of things as a function of previous measurements. This is called a recurrence.

Recurrences



Your bank account balance looks very much like what it was a month ago!

We can reliably estimate current state of things as a function of previous measurements. This is called a recurrence.

Logistic map, factorials, fibonacci sequences etc.

Solving Recurrences

- Many famous/important algorithms use the recursion.
- For example, MergeSort, QuickSort, Tree searching, Strassen's algorithm etc.
- Time Complexity of these algorithms is written as a recurrence.
- There are multiple ways to solve a recurrence.
- We will see a couple of them today.

Solving Recurrences

- Many famous/important algorithms use the recursion.
- For example, MergeSort, QuickSort, Tree searching, Strassen's algorithm etc.
- Time Complexity of these algorithms is written as a recurrence.
- There are multiple ways to solve a recurrence.
- We will see a couple of them today.

But we need some practice on summation of series and finding closed forms.

Summation and closed form

A closed form is an expression that involves very few operations ($+, -, \times, \div$ etc).

> A closed form is an expression that can be computed by applying a fixed number of familiar operations to arguments (e.g., $n(n+1)$).

Determining the closed form can make it easier to interpret.
The following expression: $2 + 4 + \ldots + 2n$ (is not closed form).

Summation and closed form

Some helpful facts for deriving closed forms:

- $\sum_{k=1}^{n} c = c(n)$ [Sum of Constant]
- $\sum_{k=m}^{n} c = c(n-m+1)$ [Sum of Constant]
- $\sum ca_k = c\sum a_k$ [Sum of Constant]
- $\sum_{k=1}^{n}(a_k - a_{k-1}) = a_n - a_0$ [Collapsing sum]
- $\sum_{k=1}^{n}(a_{k-1} - a_k) = a_0 - a_n$ [Collapsing sum]
- $\sum(a_k + b_k) = \sum(a_k) + \sum(b_k)$ [Sum of sums]
- $\sum_{k=m}^{n} a_{k+i} = \sum_{k=m+i}^{n+i} a_k$ [Shifting index]

Summation and closed form

Some well-known closed forms that everyone should know:

- $\sum(a_k x^{i+k}) = x^i \sum(a_k x^k)$
- $\sum_1^n k = \frac{(n(n+1))}{2}$
- $\sum_{k=1}^n k^2 = \frac{(n(n+1)(2n+1))}{6}$
- $\sum_{k=0}^n a^k = \frac{a^{n+1}-1}{a-1} \quad a \neq 1$
- $\sum_{k=1}^n ka^k = \frac{a-(n+1)a^{n+1}+na^{n+2}}{(a-1)^2} \quad a \neq 1$

Summation and closed form

Let's take a look at how we might convert a given summation into a closed form:

Example

Find a closed form for the following:
$3 + 7 + 11 + ... + (3 + 4n)$

Summation and closed form

Let's take a look at how we might convert a given summation into a closed form:

Example

Find a closed form for the following:
$3 + 7 + 11 + ... + (3 + 4n)$

Step 1: Get a handle on the summation

$= 3 + 7 + 11 + ... + (3 + 4n) = \sum_{k=0}^{n}(3 + 4k)$

Step 2: Try to simplify or apply any known tricks

$= 3(n + 1) + \sum_{k=0}^{n}(4k) = 3(n + 1) + \sum_{k=1}^{n}(4k)$

$= 3(n + 1) + 4\sum_{k=1}^{n} k = 3(n + 1) + 4(\frac{n(n+1)}{2})$

$= 3 + 3n + 2n^2 + 2n = 2n^2 + 5n + 3$

Summation and Closed forms

Try the following.

Example

Find a closed form for the following expression:
$\sum_{k=1}^{n}(2k+3)$

Summation and Closed forms

Try the following.

Example

Find a closed form for the following expression:
$\sum_{k=1}^{n}(2k+3)$

$\sum_{k=1}^{n}(2k+3) = 3n + 2\sum_{k=1}^{n} k$
$= 3n + 2(\frac{n(n+1)}{2}) = 3n + n(n+1) = n^2 + 4n$

Summation and Closed forms

Try the following.

Example

Find a closed form for the following sequence:
$4 + 8 + 12 + 16 + \ldots + 4n$

Summation and Closed forms

Try the following.

Example

Find a closed form for the following sequence:
$4 + 8 + 12 + 16 + ... + 4n$

$4i$ [Try to find the pattern (find the i^{th} term)]

$\sum_{i=1}^{n} 4i$ [Write the summation expression]

$= \sum_{i=1}^{n} 4i$ [Simplify- find the closed form]

$= 4 \sum_{i=1}^{n} i$

$= 4(\frac{n(n+1)}{2})$

$= 2n^2 + 2n$

Summation and Closed forms

Try the following.

Example

Find a closed form for the following (a bit challenging):
$2 + (2^2 \times 7) + (2^3 \times 14) + (2^4 \times 21) + (2^n \times 7(n-1))$

Answer

$2^i \times 7(i-1)$ [Try to find the pattern (find the i^{th} term)]

$= 2 + \sum_{i=1}^{n} 2^i \times 7(i-1)$ [Write the summation expression]

$= 2 + 7\sum_{i=1}^{n} 2^i(i-1)$ [Simplify ,take constant out (Fact 2)]

$= 2 + 7\sum_{i=1}^{n} (2^i i - 2^i)$ [Simplify ,(Fact 5)]

$= 2 + 7(\sum_{i=1}^{n} 2^i i - \sum_{i=1}^{n} 2^i)$

Summation and Closed forms

Example

Find a closed form for the following (a bit challenging):
$2 + (2^2 \times 7) + (2^3 \times 14) + (2^4 \times 21) + (2^n \times 7(n-1))$

$= 2 + 7(\sum_{i=1}^{n} 2^i i - \sum_{i=1}^{n} 2^i)$ [Simplify ,(Fact 11)]

$= 2 + 7((2 - (n+1)2^{n+1} + n2^{n+2}) - \sum_{i=1}^{n} 2^i)$ [Simplify ,(Fact 10)]

$= 2 + 7((2 - (n+1)2^{n+1} + n2^{n+2}) - (2^{n+1} - 1 - 1))$ [Simplify red term]

$= 2 + 7((2 + (n-1)2^{n+1}) - (2^{n+1} - 1 - 1))$ [Simplify blue term]

$= 2 + 7((2 + (n-1)2^{n+1}) - (2^{n+1} - 2))$

$= 2 + 7(4 + (n-2)2^{n+1})$ [Further simplification]

Summation and closed forms

Example

Let count(n) be the number of := statements executed by the following algorithm as a function of n. Find a closed form for a count(n).

```
i := 1;
while i < n
    i := i + 1;
        for j := 1 to i
            do S
        end
end
```

1 assignment op

$(n-1)$

$\times(2+3+\ldots..+n)$ ops

Summation and closed forms

Answer:

The total count of operations, count(n) is:

$$= 1 + (n-1) + (2+3+4+\ldots+n)$$

$$= 1+2+3+4+\ldots+n+(n-1) = \frac{n(n+1)}{2} + (n-1)$$

$$= \frac{n(n+1)}{2} + \frac{2(n-1)}{2} = \frac{n^2+3n-2}{2}$$

Summation and closed forms

Lets try a harder one.

Example

Let count(n) be the number of := statements executed by the following algorithm as a function of n. Find a closed form for a count(n).

```
i := 1;                          1
while i < n                      (n-1)/2
   i := i + 2;                   (i goes up by 2) ops
      for j := 1 to i            ???
        do S
      end
end
```

Summation and closed forms

Lets try a harder one.

Let's pick a specific n (i.e., n = 8) to just try to get a handle on what the heck is happening inside the loop.

```
i := 1;
while i < n
   i := i + 2;
      for j := 1 to i
         do S
      end
end
```

i	i<8?	j
1	T	1 to 3
3	T	1 to 5
5	T	1 to 7
7	T	1 to 9
9	F	

Summation and closed forms

Lets try a harder one.
Let's pick a specific n (i.e., n = 8) to just try to get a handle on what the heck is happening inside the loop. Assume $n = 2k$.

```
i := 1;
while i < n
    i := i + 2;
        for j := 1 to i
            do S
        end
end
```

i	i<n?	j	calls to S
1	T	1 to 3	3
3	T	1 to 5	5
5	T	1 to 7	7
7	T	1 to 9	9
2k-1	T	1 to 2k+1	2k+1
2k+1	F		

Summation and closed forms

So the number of calls to S looks like

Calls to S:

$$
\begin{aligned}
& 3+5+7+\ldots..+(2k+1) \\
&= \sum_{h=1}^{k}(2h+1) = \sum_{h=1}^{k}(2h) + \sum_{h=1}^{k}(1) \\
&= 2 \times \frac{k(k+1)}{2} + k \\
&= k^2 + 2k
\end{aligned}
$$

We now know the total number of calls to S. But what is $k^2 + 2k$ in terms of n ?

Summation and closed forms

So the number of calls to S looks like

Calls to S:

$$
\begin{aligned}
& 3+5+7+\ldots..+(2k+1) \\
&= \sum_{h=1}^{k}(2h+1) = \sum_{h=1}^{k}(2h) + \sum_{h=1}^{k}(1) \\
&= 2 \times \frac{k(k+1)}{2} + k \\
&= k^2 + 2k
\end{aligned}
$$

We now know the total number of calls to S. But what is $k^2 + 2k$ in terms of n ?

$$k^2 + 2k = n/2(n/2+2) = \Theta(n^2)$$

Summations and Recurrence Relation

Next :

What if sequence is given in the form of a recurrence relation (instead of individual terms) ?

$$a(n) = c_1 a(n-1) + c_2$$

How can we find the closed form of the recurrence relation a(n) ?

Recurrence Relations : Substitution method

Given a recurrence relation:
$a(n) = a(n-1) + 2n; \qquad a(0) = 1$
Find the closed form of $a(n)$.

Step1:

Work backwards to unravel and write out some terms.

$$a(n) = a(n-1) + 2n$$
$$a(n-1) = a(n-2) + 2(n-1)$$
$$a(n-2) = a(n-3) + 2(n-2)$$

Step2:

Use substitutions and plug in for the terms.

$$a(n) = a(n-1) + 2n$$
$$a(n) = a(n-2) + 2(n-1) + 2n$$
$$a(n) = a(n-3) + 2(n-2) + 2(n-1) + 2n$$

Recurrence Relations : Substitution method

Given a recurrence relation: $a(n) = a(n-1) + 2n; \qquad a(0) = 1$ Find the closed form of $a(n)$.

Step3:

Look for a pattern among the terms and visualize the terms over its entirety.

$$a(n) = a(n-3) + 2(n-2) + 2(n-1) + 2n$$

i^{th} term = 2i

Step4:

Establish the summation formula and then determine the closed form

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

So, the final closed form would be: $n^2 + n{+}1 = O(n^2)$ add 1 for initial term a(0)

Substitution method-More examples

Practice: Analyze how much work a recursive algorithm does by precisely calculating its closed form based on the following recurrence relation.

$$r(n) = 2r(n-1) + 1; \quad r(0) = 1$$

Substitution method-More examples

Step 1: Work backwards and write out some terms:

$r(n) = 2r(n-1) + 1$
$r(n-1) = 2r(n-2) + 1$
$r(n-2) = 2r(n-3) + 1$
$r(n-3) = 2r(n-4) + 1$

Step 2: Use substitutions for the terms...

$r(n) = 2r(n-1) + 1$
$= 2[2r(n-2) + 1] + 1$
$= 2^2 r(n-2) + 2 + 1$
$= 2^2[2r(n-3) + 1] + 2 + 1$
$= 2^3 r(n-3) + 2^2 + +2 + 1$

Do you see a pattern? This looks like a geometric sequence (there's a summation formula for that).

$$2^0 + 2^1 + 2^2 + 2^3 + \ldots + 2^n$$

Substitution method-Cont...

Step 4: Try to establish the summation formula and determine the closed form.

$\sum_{k=0}^{n} a^k = \frac{a^{n+1}-1}{a-1}; a \neq 1$

This leads us to a closed form of $2^{n+1} - 1$ which is $O(2^n)$.

Recurrence Relations: Cancellation Method

Another technique that is useful for solving a recurrence relation is the:

Cancellation method.

Recurrence Relations: Cancellation Method

How it works.

1. Start with the general (recurrence) equation $r(n)$ and continue writing new equations whose left side is the term involving the recurrence r on the right side of the previous equation.
2. Look for a pattern that allows us to skip ahead and write the last equation $r(0)$ on the right side.
3. Add up the equations and cancel the like terms to obtain an expression for $r(n)$.

Recurrence Relations: Cancellation Method

Example Solve the following recurrence using cancellation:

$a(0) = 1$

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

Step1:. As with substitution, we start by writing out some terms:

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

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

$a(n-2) = a(n-3) + 2(n-2)$

....

$a(n-(n-1)) = a(0) + 2(n-(n-1))$ same as $a(1) = a(0) + 2$

Recurrence Relations: Cancellation Method

Step 2: Next we add up the terms on both sides of the $=$ sign

$$a(n) + \cancel{a(n-1)} + \cancel{a(n-2)} + \cancel{a(n-3)} + \ldots + \cancel{a(1)} + \cancel{a(0)}$$

$$=$$

$$\cancel{a(n-1)} + 2n + \cancel{a(n-2)} + 2(n-1) + \cancel{a(n-3)} + 2(n-2) + \ldots + \cancel{a(0)} + 2 + a(0)$$

Step 3: Cancel the like terms to obtain the equation for $a(n)$.

$a(n) = a(0) + 2(1) + \ldots + 2(n-2) + 2(n-1) + 2n$

Recurrence Relations: Cancellation Method

Step 4: Determine the summation and closed form.

$a(n) = a(0) + 2(1) + ... + 2(n-2) + 2(n-1) + 2n$

$a(n) = 1 + 2[1 + ... + (n-2) + (n-1) + n]$

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

$$a(n) = n^2 + n + 1$$

This is an $O(n^2)$ algorithm

Recurrence Relations: Tree Method

Tree Method is a very intuitive method to deduce or guess the solution to a recurrence pictorially.

We will study and see examples of this method later once we have defined graphs and trees.