VANDERBILT UNIVERSITY School of Engineering

Discrete Structures

CS 2212

(Fall 2020)

15 – Induction and Recurrence

Sequences - Terminology

Sequence: A special type of function in which the domain is a consecutive set of integers.

index →	k	0	1	2	3	4	5	6	7	8	9	10
term →	$g(k)$	6	13	23	33	54	83	118	156	210	282	350

$$g(k) = g_k$$

Finite sequence: A sequence with a finite domain.

Infinite sequence: A sequence with an infinite domain.

- Initial / Final index.
- Initial / Final term.

Sequences - Terminology

A sequence can be specified by an **explicit formula** showing how the value of term a_k depends on k.

Example:

k	0	1	2	3	4	5	6	7	8	9	10
a_k	1	2	4	8	16	32	64	128	256	512	1024

Here, $a_k = 2^k$ for $k \geq 1$.

Sequences - Terminology

Increasing: For *every* two consecutive indices, k and $k+1$, in the domain

$$a_k < a_{k+1}.$$

Non-decreasing: For *every* two consecutive indices, k and $k+1$, in the domain,

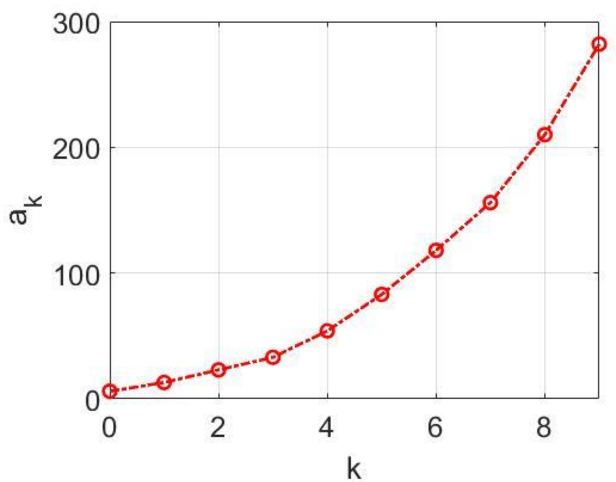
$$a_k \le a_{k+1}.$$

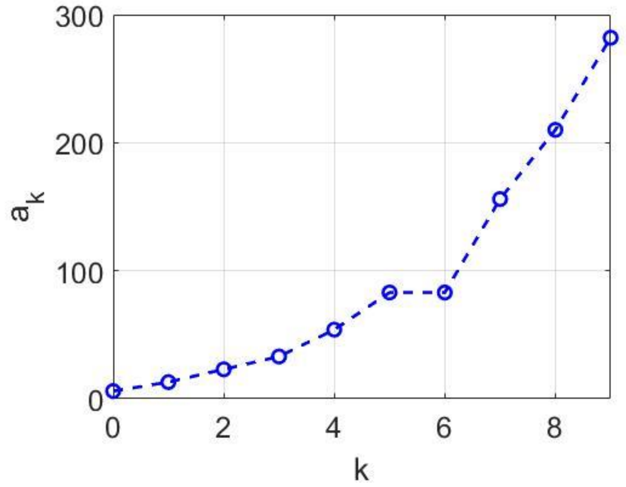
k	0	1	2	3	4	5	6	7	8	9
a_k	6	13	23	33	54	83	118	156	210	282

Increasing

k	0	1	2	3	4	5	6	7	8	9
a_k	6	13	23	33	54	**83**	**83**	156	210	282

Non-decreasing





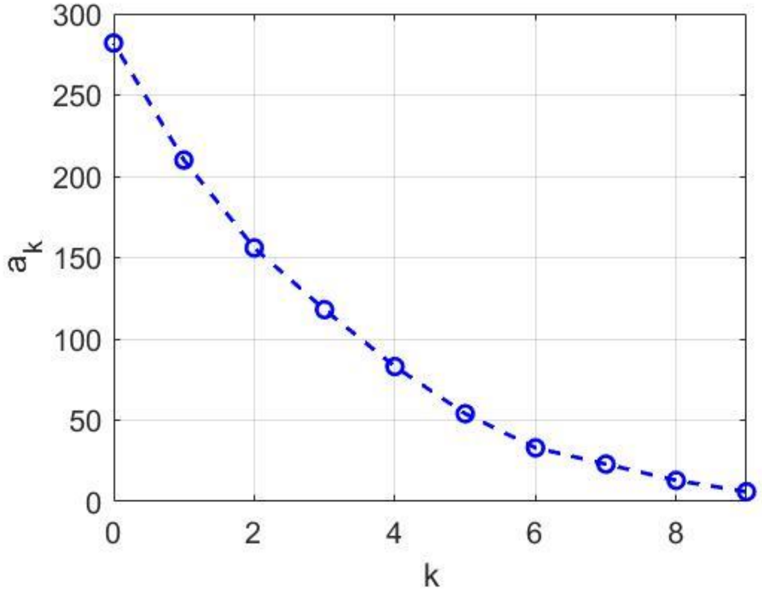
Sequences - Terminology

Decreasing: For *every* two consecutive indices, k and $k+1$, in the domain

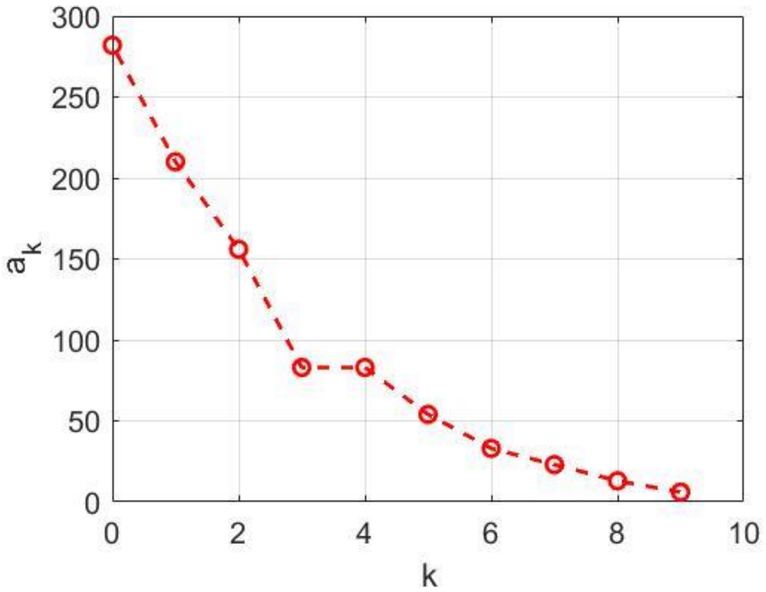
$$a_k > a_{k+1}.$$

Non-increasing: For *every* two consecutive indices, k and $k+1$, in the domain,

$$a_k \geq a_{k+1}.$$

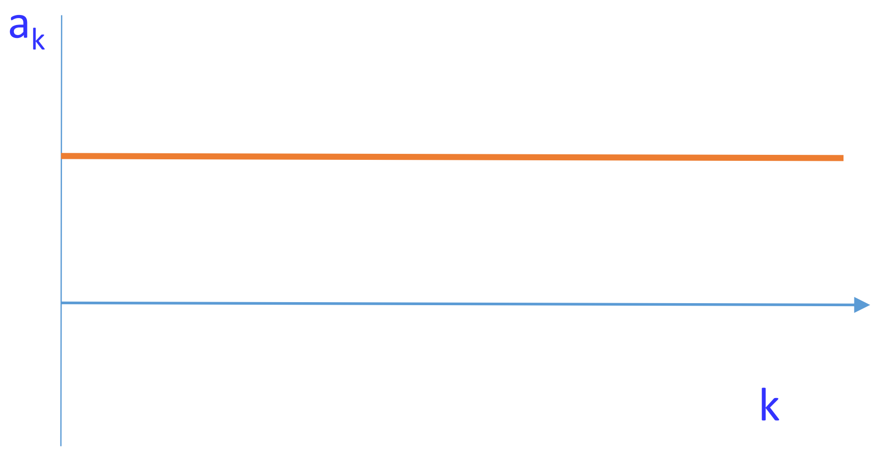


Decreasing



Non-increasing

Sequences - Terminology



Increasing 

Non-decreasing 

Decreasing 

Non-increasing 

Arithmetic Sequences

A sequence of real numbers where each term after the initial term is found by taking the previous term and **adding** a fixed number called the **common difference**.

a_0 a_1 a_2 a_3 a_4 …

Arithmetic Sequences

A sequence of real numbers where each term after the initial term is found by taking the previous term and **adding** a fixed number called the **common difference**.

a_0	a_1	a_2	a_3	a_4	…
a_0					

Arithmetic Sequences

A sequence of real numbers where each term after the initial term is found by taking the previous term and **adding** a fixed number called the **common difference**.

a_0	a_1	a_2	a_3	a_4	…
a_0	$(a_0 + d)$				

Arithmetic Sequences

A sequence of real numbers where each term after the initial term is found by taking the previous term and **adding** a fixed number called the **common difference**.

a_0	a_1	a_2	a_3	a_4	…
a_0	$(a_0 + d)$	$(a_1 + d)$			

Arithmetic Sequences

A sequence of real numbers where each term after the initial term is found by taking the previous term and **adding** a fixed number called the **common difference**.

a_0	a_1	a_2	a_3	a_4	…
a_0	$(a_0 + d)$	$(a_1 + d)$	$(a_2 + d)$		

Arithmetic Sequences

A sequence of real numbers where each term after the initial term is found by taking the previous term and **adding** a fixed number called the **common difference**.

a_0	a_1	a_2	a_3	a_4	…
a_0	$(a_0 + d)$	$(a_1 + d)$	$(a_2 + d)$	$(a_3 + d)$	…

- Here, $a_k - a_{k-1} = d$, for all $k > 0$.
- Note that $a_k = a_0 + kd$
- Observe a "linear" growth.

Arithmetic Sequences

a_0	a_1	a_2	a_3	a_4	…
a_0	$(a_0 + d)$	$(a_1 + d)$	$(a_2 + d)$	$(a_3 + d)$	…

Example:

- Suppose a person inherits a collection of 500 baseball cards and decides to continue growing the collection at a rate of 10 additional cards each week.
- A sequence of cards at the end of each week is an arithmetic sequence.

Geometric Sequences

A sequence where each term after the initial term is found by taking the previous term and **multiplying** by a fixed number called the **common ratio**.

a_0	a_1	a_2	a_3	…	a_k
a_0	$r \times a_0$	$r \times a_1$	$r \times a_2$	…	$r \times a_{k-1}$

- Here, $\frac{a_k}{a_{k-1}} = r$, for all k.
- Note that $a_k = r^k a_0$
- Observe a "non-linear" growth.

Example: Money in a bank account earning a fixed rate of interest can be expressed as a geometric sequence.

Recurrence Relation

Recurrence Relation: A rule that defines a term a_n as a *function of previous terms* in the sequence is called a recurrence relation.

Arithmetic sequence

$a_0 = a$ (initial value)
$a_n = d + a_{n-1}$ for $n \geq 1$ (recurrence relation)

Initial value = a. Common difference = d.

Geometric sequence

$a_0 = a$ (initial value)
$a_n = r \cdot a_{n-1}$ for $n \geq 1$ (recurrence relation)

Initial value = a. Common ratio = r.

Recurrence Relation

Sometimes, we need multiple previous values to obtain the current term in the sequence. For instance,

$f_0 = 0$
$f_1 = 1$] initial values

$f_2 = f_1 + f_0 = 1 + 0 = 1$

$f_3 = f_2 + f_1 = 1 + 1 = 2$

$f_4 = f_3 + f_2 = 2 + 1 = 3$

Here,

$$f_k = f_{k-1} + f_{k-2}$$

Such a sequence is known as Fibonacci sequence.

- Many applications: coding, optimization, algorithms analysis, arts etc.
- Golden ratio
- Miles to Kilometers (fun application)

Recurrence Relation

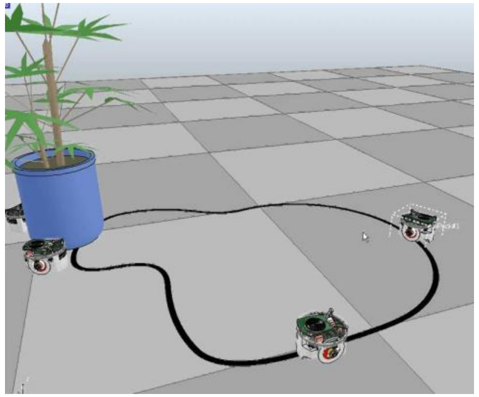
Other applications of recurrence relations include modelling of (discrete time) **dynamical systems**.

Difference equations are used to model dynamics of such systems.

Simplest difference equation has the form.

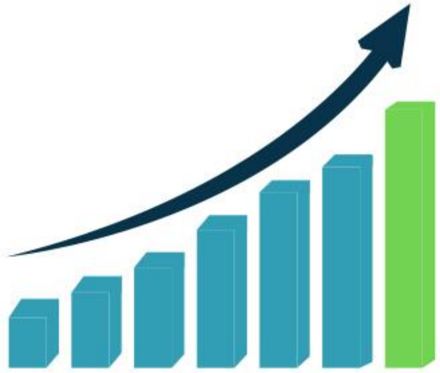
$$x_{n+1} = a\, x_n$$

← Recurrence relation



Robot dynamics

Population growth



Economic models

Recurrence Relation

Example: Describe the scenario below as a *recurrence relation*.

- The salary of an employee in year n is s_n
- The employee receives a \$1000 bonus in December.
- The salary for the subsequent year is 5% more than the income received by the employee in the previous year (including salary and bonus).

Recurrence Relation

Answer:

- The income earned in year n - 1 is $(s_{n-1} + 1000)$.
- Thus, the increase in salary is 5% of $(s_{n-1} + 1000)$.
- If a quantity increases by 5%, then the quantity becomes 1.05 times larger.
- Therefore the salary in year n can be described by the recurrence relation $1.05(s_{n-1} + 1000)$.

Sequences and Summations

Summation notation is used to express the sum of terms in a numerical sequence

Summation form: $$\sum_{i=1}^{4} i^2$$

Expanded form: $1^2 + 2^2 + 3^2 + 4^2$

Always use parentheses as necessary to indicate which terms are included in the summation as in $\sum_{i=1}^{n}(i + 1)$

Summations: Tips and Tricks

- Sometimes, it helps to "manipulate" the summation expression and make things easy, for instance, to compute the exact value of the expression, or to simplify the form of the expression.
- We will take a look at some tricks and tips to do so.
- Before that, lets enjoy a **Gauss story**.

$$\sum_{i=1}^{100} i = 1 + 2 + 3 + \ldots + 100 = ?$$

$$= (1+100) + (2+99) + (3+98) + \ldots + (50 + 51)$$

$$= \underbrace{101 + 101 + 101 + \ldots + 101}_{\text{50 terms}}$$

$$\sum_{j=1}^{50} (101) = 50\,(101) = 5050$$

1777 - 1855

Summations: Tips and Tricks

Tip #1: Pulling out a final term from a summations

- In working with summations, it is sometimes useful to be able to *pull out* (or *add in*) a *final term* to a summation.
- This is often done so we can use a specific closed formula (*e.g.*, $\frac{n(n+1)}{2}$)
- For $n > m$, we have the following trick:

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

Summations: Tips and Tricks

Try the following:

$$\sum_{j=0}^{n} (2j) \;=\; ??$$

$$\sum_{j=0}^{n+2} \left(2^{j-1}\right) \;=\; ??$$

Answers:

$$\sum_{j=0}^{n} (2j) \;=\; \sum_{j=0}^{n-1} (2j) \;+\; 2n$$

$$\sum_{j=0}^{n+2} \left(2^{j-1}\right) \;=\; \sum_{j=0}^{n+1} \left(2^{j-1}\right) \;+\; 2^{n+1}$$

Summations: Tips and Tricks

Tip #2: Change of variables in summations

- The variable used for the index in a summation is internal to the sum and can be *replaced* with any other variable name (i.e., the value of i, j, k).
- Substitutions can be done for the index variable and will require that both the *upper and lower limit be adjusted*.

Summations: Tips and Tricks

Example: Let's substitute i for $k+2$ in the summation below:

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

We want to have an expression of the form

$$\sum_{i=?}^{?} i^3$$

What should be the correct upper and lower limits (in terms of i)?

Lower limit: When $k = 1$, what should be the value of i?

We have selected, $i = k + 2$

So, $k = 1 \Rightarrow i = 3$

Summations: Tips and Tricks

Example: Let's substitute i for $k+2$ in the summation below:

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

Similarly,

Upper limit: When $k = n$, what should be the value of i ?

We have selected, $i = k + 2$

So, $k = n \Rightarrow i = n + 2$

So, we get

$$\sum_{i=3}^{n+2} i^3$$

Summations and Closed Forms

When analyzing an algorithm we often run across coding that involves some type of loop and we need to "count" the number of times the loop is executed.

Sometimes things are pretty *easy* to count …

```
FOR i = 1 to n
     sum = sum + i   //do this statement n times
End FOR
```

Summations and Closed Forms

When analyzing an algorithm we often run across coding that involves some type of loop and we need to "count" the number of times the loop is executed.

Sometimes things are pretty *not so easy* to count …

```
i = 1;
WHILE (i < (n-3)) do
     i = i + 1;
     FOR j = 1 to i
          do stuff  // do this statement ??? times
     End FOR
End WHILE
```

Summations and Closed Forms

When it is not easy to count, we proceed by attempting to find a **closed form** to assist in counting operations.

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 than trying to count the number of operations in the following expression: $2 + 4 + \ldots + 2n$ (not closed form).

Summations and Closed Forms

Some helpful facts for deriving closed forms:

1. $$\sum_{k=m}^{n} c = c(n-m+1)$$ [Sum of constant]

2. $$\sum c\, a_k = c \sum a_k$$ [Sum of constant]

3. $$\sum_{k=1}^{n} (a_k - a_{k-1}) = a_n - a_0$$ [Collapsing sum]

4. $$\sum_{k=1}^{n} (a_{k-1} - a_k) = a_0 - a_n$$ [Collapsing sum]

Summations and Closed Forms

5. $\sum (a_k + b_k) = \sum a_k + \sum b_k$ [Sum of sums]

6. $\sum_{k=m}^{n} a_{k+i} = \sum_{k=m+i}^{n+i} a_k$ [Shifting index]

7. $\sum (a_k \, x^{i+k}) = x^i \sum \left(a_k \, x^k\right)$

8. $\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$

Summations and Closed Forms

9. $$\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$$

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

11. $$\sum_{k=1}^{n} k\, a^k = \frac{a - (n+1)a^{n+1} + na^{n+2}}{(a-1)^2}; \quad a \neq 1$$

Summations and Closed Forms

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:

$$3 + 7 + 11 + \ldots + (3 + 4n)$$

Step 1: Get a handle on the summation

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

Step 2: Try to simplify or apply any known tricks

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

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

$= 3 + 3n + 2n^2 + 2n =$ $\mathbf{2n^2 + 5n + 3}$

Summations and Closed Forms

Try the following.

Example: Find a closed form for the following expression:

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

Answer:

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

$$= 3n + 2\left(\frac{n(n+1)}{2}\right) = 3n + n(n+1) = \boxed{\boldsymbol{n^2 + 4n}}$$

Summations and Closed Forms

Example: Find a closed form for the following sequence:

$$4 + 8 + 12 + 16 + \ldots + 4n$$

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

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

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

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

Summations and Closed Forms

Example: Find a closed form for the following (a bit challenging):

$$2 + (2^2 \times 7) + (2^3 \times 14) + (2^4 \times 21) + (2^n \times 7(n-1))$$

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)]

Summations and Closed Forms

Example: Find a closed form for the following (a bit challenging):

$$2 + (2^2 \times 7) + (2^3 \times 14) + (2^4 \times 21) + (2^n \times 7(n-1))$$

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)]

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

Summations and Closed Forms

Example: Find a closed form for the following (a bit challenging):

$$2 + (2^2 \times 7) + (2^3 \times 14) + (2^4 \times 21) + (2^n \times 7(n-1))$$

Answer: (continued)

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

$$= 2 + 7\left(\left(2 - (n+1)2^{n+1} + n2^{n+2}\right) - \sum_{i=1}^{n} 2^i\right)$$ [Simplify – (*Fact* 11)]

$$= 2 + 7\left(\left(2 - (n+1)2^{n+1} + n2^{n+2}\right) - \left(2^{n+1} - 1 - 1\right)\right)$$

[Simplify – (*Fact* 10)]

Summations and Closed Forms

Example: Find a closed form for the following (a bit challenging):

$$2 + (2^2 \times 7) + (2^3 \times 14) + (2^4 \times 21) + (2^n \times 7(n-1))$$

Answer: (continued)

$$= 2 + 7\left(\left(2 - (n+1)2^{n+1} + n2^{n+2}\right) - \left(2^{n+1} - 1 - 1\right)\right)$$

$$= 2 + 7\left(2 + (n-1)2^{n+1} - \left(2^{n+1} - 1 - 1\right)\right)$$ [Simplify green term]

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

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

Summations 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

$c_1\ (n - 1)$

$c_2 \times (2 + 3 + \ldots + n)$ ops

Summations and Closed Forms

Answer:

We ignore constants c_1 and c_2.

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

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

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

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

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

(1)

$(n - 1)/2$

(**i** goes up by 2)

???

Summations and Closed Forms

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 do
    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	

Summations and Closed Forms

Let's generalize it now for any n.

```
i := 1;
while i < n do
    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		

Summations and Closed Forms

So the number of calls to S looks like

Calls to S:

$$3 + 5 + 7 + \cdots + (2k+1)$$

$$= \sum_{h=1}^{k}(2h+1) \qquad = \sum_{h=1}^{k}(2h) + \sum_{h=1}^{k}(1)$$

$$= 2 \times \frac{k(k+1)}{2} + k$$

$$= k^2 + 2k$$

We now know the total number of calls to S. But what is $k^2 + 2k$ in terms of n ? That's really what we need to know.

Summations and Closed Forms

- **Question on the table:** What is $(k^2 + 2k)$ in terms of n?
- We can plug that result back into the closed form on the previous slide for k to get the total count of calls to S.

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

$$(2k-1) < n \leq (2k+1)$$

$$(2k-2) < (n-1) \leq (2k)$$

$$(k-1) < \frac{n-1}{2} \leq k$$

Summations and Closed Forms

Summary:

Given a sequence (**terms**): $a_1, a_2, a_3, \ldots a_n$

Find sequence sum: $a_1 + a_2 + a_3 + \ldots + a_n$

$a_1 + a_2 + a_3 + \ldots + a_n$ — [Expanded form]

↓ Find pattern – i^{th} term (a_i)

$$\sum_{i=1}^{n} a_i$$ — [Summation]

↓ Simplify

Closed form in terms of n — [Closed form]