04/07/2025: Algorithm Efficiency

CSCI 246: Discrete Structures

Textbook reference: Sec 11.3, Epp

Graded Quiz Pickup

Quizzes are in the front of the room, grouped into four bins (A-G, H-L, M-R, S-Z) by last name. The quizzes are upside down with your last name on the back. Come find yours before, during, or after class. Only turn the quiz over if it's yours.

Today's Agenda

- Reading quiz (5 mins)
- Mini-lecture ($\approx$ 15 mins)
- Group exercises ($\approx$ 20 mins)

Feedback on Friday’s Quizzes

Problem Quiz Scores

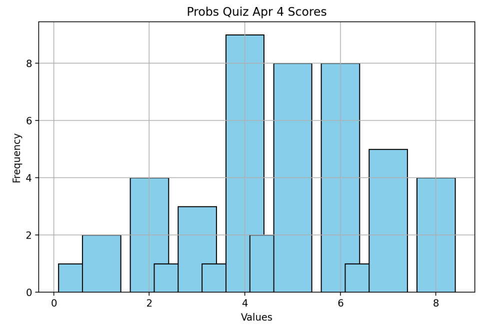

Figure 1: Median Score = 5/8 (62.5%)

Grading Rubric:

1. (4 points.) 2 points for $P(A|B)$ and 2 for $P(B|A)$. Must show work for full credit.
2. (4 points.) 1 point for correct answer, 1 point for stating correctly one of the 3 characterizations of independence, 2 points for correctly determining the probabilities needed for that characterization.

Reading Quiz Scores (Extra Credit)

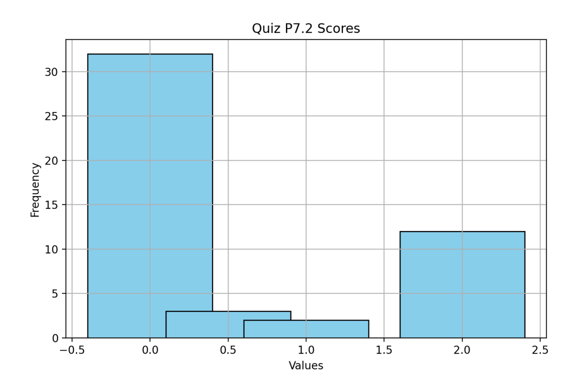

Figure 2: Median Score = 0 points extra credit

Grading Rubric: 2 points extra credit for perfect answer.

Today's reading quiz

Reading Quiz (Algorithm Efficiency)

Assume n is a positive integer and consider the following algorithm segment:

$$p := 0, x := 2$$

$$\textbf{for } i := 2 \textbf{ to } n$$

$$\quad p := (p + i) \cdot x$$

$$\textbf{next } i$$

1. What is the actual number of elementary operations that are performed when this algorithm segment is executed? Justify your answer.
2. What is the order for this algorithm segment? Justify your answer.

Review of Problems Quiz

Thoughts On Algorithm Efficiency

Theorem: Sum of the first n natural numbers

For every natural number $n \geq 1$,

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

Theorem: Sum of the first n natural numbers

For every natural number $n \geq 1$,

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

Story: Gauss' solution from elementary school

1	2	3	4	5	6	7	8	9	10
10	9	8	7	6	5	4	3	2	1
11	11	11	11	11	11	11	11	11	11

Theorem: Sum of the first n natural numbers

For every natural number $n \geq 1$,

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

Story: Gauss' solution from elementary school

$$\begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ \hline 11 & 11 & 11 & 11 & 11 & 11 & 11 & 11 & 11 & 11 \end{array}$$

Remark. Recall that we can use summation notation to express the left-hand side more succinctly; in that case, we write the theorem as:

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

Poll. How does the theorem apply to algorithm efficiency?

Poll. How does the theorem apply to algorithm efficiency?

Solution. Consider an algorithm segment with a nested for loop, such as the one below.

An algorithm segment with a nested for loop

```
for i = 1 to n:
    for j = 1 to i:
        print "YOLO"
```

Poll. How does the theorem apply to algorithm efficiency?

Solution. Consider an algorithm segment with a nested for loop, such as the one below.

An algorithm segment with a nested for loop

```
for i = 1 to n:
    for j = 1 to i:
        print "YOLO"
```

The number of times the inner loop is run is

$$1 + 2 + 3 + 4 + \ldots + n$$

We need this quantity to compute algorithmic efficiency, since the number of elementary operations performed by an algorithm segment is given by

$$\#\text{elem. ops} = \#\text{inner loops} \times \#\text{elem. ops. per inner loop}$$

Remark: Big O is not worst case

Method for computing # elem. operations		**Method for expressing order of algorithm**
Worst Case	$\times$	Big O
Best Case		Big Theta
Average Case		Big Omega

Group exercises

aaron.loomis: 5
adam.wyszynski: 6
alexander.goetz: 18
alexander.knutson: 6
anthony.mann: 20
blake.leone: 16
bridger.voss: 14
caitlin.hermanson: 12
cameron.wittrock: 3
carsten.brooks: 1
carver.wambold: 4
colter.huber: 20
conner.reed1: 7
connor.mizner: 9
connor.yetter: 1
derek.price4: 7
devon.maurer: 17
emmeri.grooms: 4
erik.moore3: 9
ethan.johnson18: 12
evan.barth: 7
evan.schoening: 21
griffin.short: 8
jack.fry: 2
jacob.ketola: 15
jacob.ruiz1: 5
jacob.shepherd1: 14
jada.zorn: 21
jakob.kominsky: 18
james.brubaker: 15
jeremiah.mackey: 19
jett.girard: 2
john.fotheringham: 6
jonas.zeiler: 4
joseph.mergenthaler: 11
joseph.triem: 17
julia.larsen: 2
justice.mosso: 13
kaden.price: 15
lucas.jones6: 10
luka.derry: 3
luke.donaldson1: 16
lynsey.read: 17
mason.barnocky: 16
matthew.nagel: 8
micaylyn.parker: 10
michael.oswald: 13
nolan.scott1: 18
owen.obrien: 21
pendleton.johnston: 13
peter.buckley1: 9
reid.pickert: 8
ryan.barrett2: 14
samuel.hemmen: 12
samuel.mosier: 11
samuel.rollins: 3
sarah.periolat: 19
timothy.true: 11
tristan.nogacki: 1
tyler.broesel: 20
william.elder1: 5
yebin.wallace: 19
zeke.baumann: 10

Group exercises

For each of the algorithm segments below, assume *n* is a positive integer.

a. Compute the actual number of elementary operations (additions, subtractions, multiplications, divisions, and comparisons) that are performed when the algorithm segment is executed. For simplicity, however, count only comparisons that occur within the body of the **for-next** loops; ignore those required to determine when the **for-next** loops should terminate.

b. Use the theorem on polynomial orders to find an order for the algorithm segment.

Segment 1

for $i := 3$ **to** $n-1$
    $a := 3 \cdot n + 2 \cdot i - 1$
next i

Segment 2

for $k := 1$ **to** $n-1$
    for $j := 1$ **to** $k+1$
        $x := a[k] + b[j]$
    next j
next k

Segment 3

for $i := 1$ **to** n
    for $j := 1$ **to** $\lfloor (i+1)/2 \rfloor$
        $a := (n-i) \cdot (n-j)$
    next j
next i

Solution for algorithm segment #1

$$\begin{array}{l} \textbf{for } i := 3 \textbf{ to } n-1 \\ \quad a := 3 \cdot n + 2 \cdot i - 1 \\ \textbf{next } i \end{array}$$

Solution Summary.

Step 1	**# its.**	$n - 3$
Step 2	**# el ops. per it.**	4
Step 3	**# el ops.**	$4n - 12$
Step 4	**Order**	$\Theta(n)$

Solution Details.

Step 1	Since we skip the first two iterates, do $(n-1) - 2 = n - 3$.
Step 2	Two multiplications, one subtraction, one addition.
Step 3	Multiply Steps 1 and 2.
Step 4	By theorem on polynomial orders.

Solution for algorithm segment #2

for $k := 1$ **to** $n-1$
    for $j := 1$ **to** $k+1$
        $x := a[k] + b[j]$
    next j
next k

Solution Summary.

Step 1	**# its.**	$\frac{n^2}{2} - \frac{n}{2} - 1$
Step 2	**# el ops. per it.**	1
Step 3	**# el ops.**	$\frac{n^2}{2} - \frac{n}{2} - 1$
Step 4	**Order**	$\Theta(n^2)$

Solution Details.

Step 1	We unfold the nested loop into a single loop by focusing only on the inner loop. The inner loop does $2 + \ldots + n = \frac{n(n+1)}{2} - 1$ iterations. Here we use Gauss' formula for summing the first n natural numbers.
Step 2	One addition.
Step 3	Multiply Steps 1 and 2.
Step 4	By theorem on polynomial orders.

Solution for algorithm segment #3

Solution Summary.

```
for i := 1 to n
    for j := 1 to ⌊(i+1)/2⌋
        a := (n − i)·(n − j)
    next j
next i
```

Step 1	**# its.**	$\begin{cases} \frac{n^2}{4}+\frac{n}{2}+\frac{1}{4}, & n \text{ odd} \\ \frac{n^2}{4}+\frac{n}{2}, & n \text{ even} \end{cases}$
Step 2	**# el ops. per it.**	3
Step 3	**# el ops.**	$\begin{cases} 3\left(\frac{n^2}{4}+\frac{n}{2}+\frac{1}{4}\right), & n \text{ odd} \\ 3\left(\frac{n^2}{4}+\frac{n}{2}\right), & n \text{ even} \end{cases}$
Step 4	**Order**	$\Theta(n^2)$

Solution Details. How many times is the inner loop run? Depends if n is odd or even.

$$\texttt{\# its } (n \texttt{ odd}) = 1+1+2+2+\ldots+\frac{n-1}{2}+\frac{n-1}{2}+\frac{n+1}{2}$$

$$= 2\sum_{k=1}^{\frac{n-1}{2}} k + \frac{n+1}{2} \stackrel{*}{=} \cancel{2}\frac{1}{\cancel{2}}\left(\frac{n-1}{2}\right)\left(\frac{n+1}{2}\right)+\frac{n+1}{2} = \frac{n^2}{4}+\frac{n}{2}+\frac{1}{4}$$

$$\texttt{\# its } (n \texttt{ even}) = 1+1+2+2+\ldots+\frac{n}{2}+\frac{n}{2}$$

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

In the equations marked *, we use Gauss' formula for summing the first n natural numbers.