# DES227: Algorithms Design
# Final Mock Exam

curated by The Peanuts

Name............ID............Section............Seat No............

**Conditions:** Semi-Closed Book (A4 Both Sides)

**Directions:**

1. This exam has 14 pages (including this page).
2. Calculators (Casio 991 Series) are allowed.
3. Any attempt to bring a quantum computer will be considered... impressive, but still cheating.
4. Write your name clearly at the top of each page.
5. Stay calm. Breathe in. Breathe through. Breathe deep. Breathe out.
6. No brute-force cheating. We will prune you with a branch-and-bound penalty.
7. May your answers be correct, your logic flawless, and your pencil always sharp.

*For solution, click here.*

## Problem 1

Multiply $(100)_2$ with $(01)_2$ using the Recursive-Multiply algorithm. Assume the base case is when both $x$ and $y$ are 2-digit binary numbers. Show your complete calculation steps.

Multiply $100_2$ with $01_2 = 4_{10}$

$x = 100_2$ $\quad$ $y = 0001$

$x_1 = 01$ $\quad$ $x_0 = 00$ $\quad$ $y_1 = 00$ $\quad$ $y_0 = 01$

[illegible] $xy = x_1 y_1 \cdot 2^n + (P - x_1 y_1 - x_0 y_0) \cdot 2^{n/2} + x_0 y_0$

$P_1 \cdot 2^n + (P_3 - P_1 - P_2) \cdot 2^{n/2} + P_2$

$P_1 = x_1 y_1 = (01)(00) = 00_2$

$P_2 = x_0 y_0 = (00)(01) = 00_2$

$P_3 = (x_1 + x_0) \cdot (y_1 + y_0) = (01+00) \cdot (00+01) = 01$

$00_2 \cdot 2^4 + (01_2 - 00_2 - 00_2) \cdot 2^2 + 00_2$

$(01_2) \cdot 2^2$

$100_2 + 00_2$

$100_2 = 4$

=

# Problem 2

Consider the array: $[5, 2, 8, 1, 9, 3, 7, 4]$

Apply the divide-and-conquer approach (Sort-and-Count algorithm) to count the number of inversions in this array. Show all steps of the recursive calls and how the "across-half" inversions are counted during the merge steps.

A : [5, 2, 8, 1]   B : [9, 3, 7, 4]

A
Left [5, 2] → 1 invertion ; sort [2,5]
Right [8, 1] → 1 invertion ; sort [1,8]
merge

1 2 5 8
2 5 1 8

2 invertion

total : 4 invertion

sorted : [1 2 5 8]

---

B
Left [9 3] → 1 invertion ; sort [3 9]
Right [7 4] → 1 invertion ; sort [4 7]
Merge

3 4 7 9
3 9 4 7

total : 4 invertion

sorted : [3 4 7 9]

---

Merge - Count [1, 2, 5, 8] and [3 4 7 9]

| 1 | 2 | 3 | 4 | 5 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|

count : ~~0~~ → ~~1~~ → ~~4~~ → 5

∴ 5 invertion

1 2 3 4 5 7 8 9
1 2 5 8 3 4 7 9

(5, 3) (5, 4) → 2 invertion
(8, 3) (8, 4) (8, 7) → 3 invertion

## Problem 3

Use dynamic programming to solve the following coin-changing problem:

Given coins of denominations {1, 3, 4, 5} and an amount of 7, find the minimum number of coins needed to make this amount. Show your work by filling in the complete DP table and explaining your approach.

$d_1 = 1$ $d_2 = 3$ $d_3 = 4$ $d_4 = 5$ $N = 7$

| | | 0 \$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|---|
| $d_1$ | 1 \$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| $d_2$ | 3 \$ | 0 | 1 | 2 | 1 | 2 | 3 | 2 | 3 |
| $d_3$ | 4 \$ | 0 | 1 | 2 | 1 | 1 | 2 | 2 | 2 |
| $d_4$ | 5 \$ | 0 | 1 | 2 | 1 | 1 | 1 | 2 | 2 |

4 \$ + 3 \$

explain ?

from this formular $c[i,j]$ problem is min $\{c[i-1, j], 1+c[i, j-d_i]\}$

you will see each block it fill by this formular

# Problem 4

Solve the following 0/1 knapsack problem:

| item | weight | value |
|---|---|---|
| 1 | 3 | \$25 |
| 2 | 2 | \$20 |
| 3 | 4 | \$40 |
| 4 | 5 | \$50 |

$W = 7$

(a) Solve this problem using the dynamic programming approach. Show the complete DP table.

| i\j | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 25 | 25 | 25 | 25 | 25 |
| 2 | 0 | 0 | 20 | 25 | 25 | 45 | 45 | 45 |
| 3 | 0 | 0 | 20 | 25 | 40 | 45 | 60 | 60 |
| 4 | 0 | 0 | 20 | 25 | 40 | 45 | 60 | 70 |

$V[i,j] = \max\{ V[i-1,j], v_i + V[i-1, j-w_i] \}$

$V[1,3] = \max\{ 0, 25 + V[0, 3-3] \} = 25$

$V[2,2] = \max\{ 0, 20 + V[1, 2-2] \} = 20$ | $V[2,3] = \max\{25, 20 + V[1, 3-2]\} = 25$

$V[2,7] = \max\{ 0, 20 + V[1, 7-2] \} = 45$

$V[2,5] = \max\{ 0, 20 + V[1,3] \} = 45$

$V[2,4] = \max\{ 0, 20 + V[1,2] \} = 20$

20/20 ✓

$V[3,3] = \max\{ V[2,3], 40 + V[2, 3-4] \} = \{20, 0\} = 20$

$V[3,4] = \max\{ V[2,4], 40 + V[2, 4-4] \} = \{20, 40 + 0\} = 40$

$V[3,7] = \max\{ V[2,7], 40 + V[2, 7-4] \} = \{45, 40 + 20\} = 60$

$V[3,5] = \max\{ V[2,5], 40 + V[2, 5-4] \} = \{45, 40 + 0\} = 45$

$V[4,4] \max\{ V[3,4], 50 + V[3, 4-5] \} = \{40, 0\}$

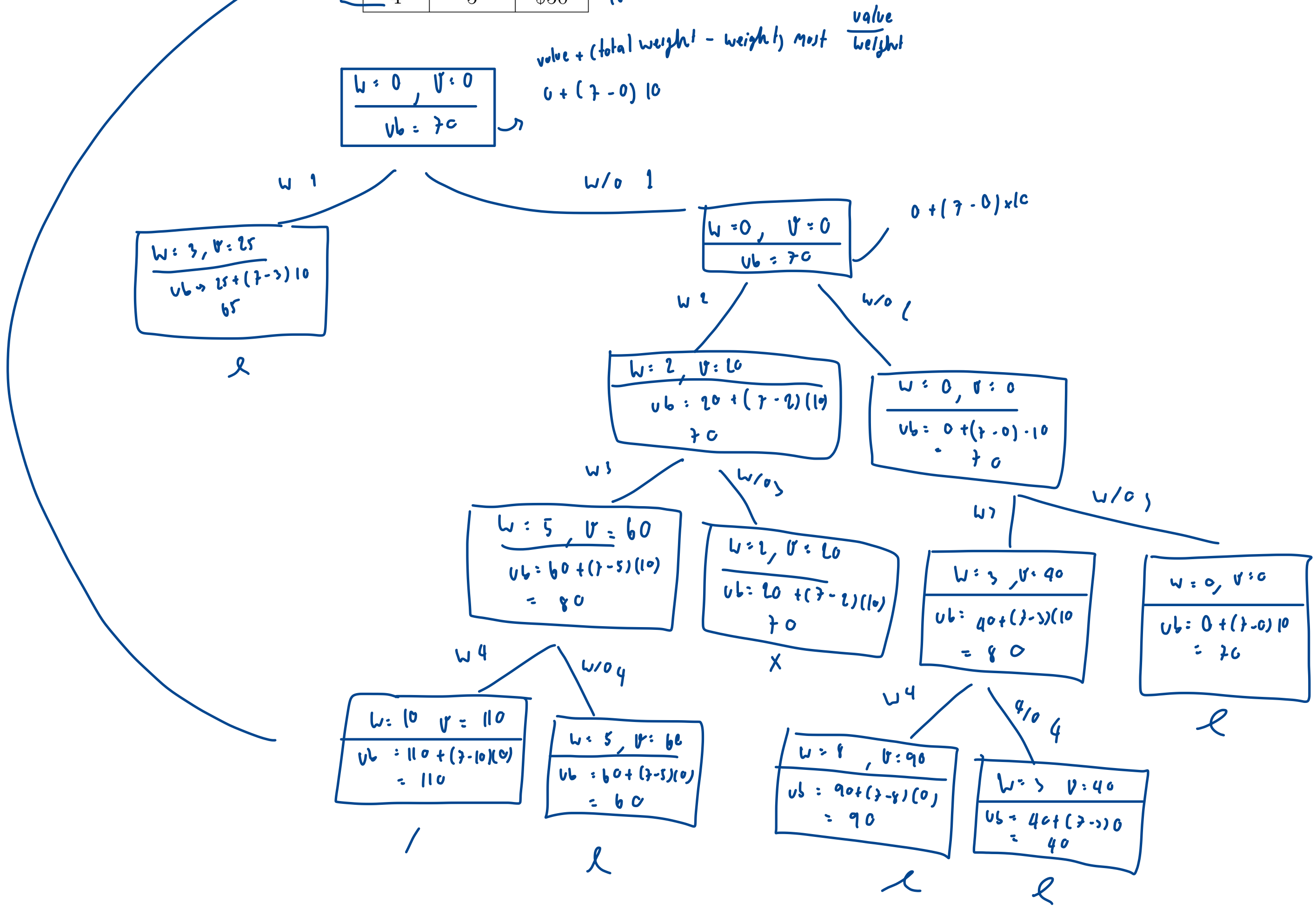
$V[4,7] \max\{ V[3,7]\ \ 50 + V[3, 7-5] \} = \{60, 50 + 20\}$

(b) Solve the same problem using the branch-and-bound algorithm. Draw the state space tree and show the calculation of upper bounds at each node.

Solve the following 0/1 knapsack problem:

| item | weight | value | $\frac{v}{w}$ |
|---|---|---|---|
| 1 | 3 | \$25 | 8.33 |
| 2 | 2 | \$20 | 10 |
| 3 | 4 | \$40 | 10 |
| 4 | 5 | \$50 | 10 |

$W = 7$




∴ w/ item 2, 3, 4

## Problem 5

Complete the given table $m[i, j]$ to solve the matrix chain multiplication problem by dynamic programming. There are 4 matrices with dimensions as follows: $M_1 : 5 \times 10$, $M_2 : 10 \times 3$, $M_3 : 3 \times 12$, and $M_4 : 12 \times 8$.

| m[i,j] | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 150 | 330 | 558 (2) |
| 2 | | 0 | 360 | 528 |
| 3 | | | 0 | 288 |
| 4 | | | | 0 |

After completing the table, determine the optimal parenthesization for multiplying these matrices.

$M_1 \cdot M_2 \cdot M_3 \cdot M_4$
5x10 10x3 3x12 12·8

$\min\{M[i,k] + m[k+1,j] + N * N * N\}$

m[1,2]
$A_1 \cdot A_2$
5x10 10x3
150

m[2,3]
$A_2 \cdot A_3$
10x3 3x12
360

m[3,4]
$A_3 \cdot A_4$
3x12 12x8
288

m[1,3] → $1 \le k < 3$

$A_1(A_2 \cdot A_3)$
5x10 10x3 3x12
m[1,1] + m[2,3] + 5x10x12
0 + 360 + 600
960

k = 2
$(A_1 \cdot A_2)A_3$
5x10 10x3 3x12
m[1,2] + m[3,3] + 5x3x12
150 + 0 + 180
330

m[2,4] → $2 \le k < 4$
k: 2, 3
k = 2
$A_2(A_3 \cdot A_4)$
10x3 3x12 12x8
m[2,2] + m[3,4] + 10x3x8
0 + 288 + 240
528

$(A_2 \cdot A_3)A_4$
10x3 3x12 12x8
m[2,3] + m[4,4] + 10x12x8
360 + 0 + 960
1320

m[1,4] ← $1 \le k < 4$
k: 1, 2, 3

$A_1\ A_2\ A_3\ A_4$
5x10 10x3 3x12 12x8
k = 2

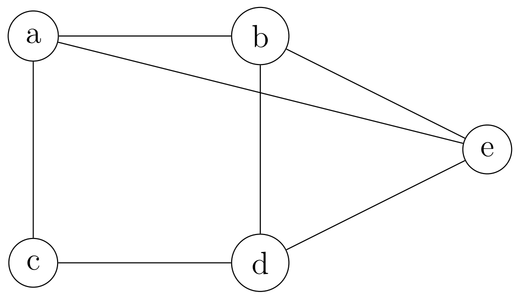
$(A_1 A_2)(A_3 A_4)$

#

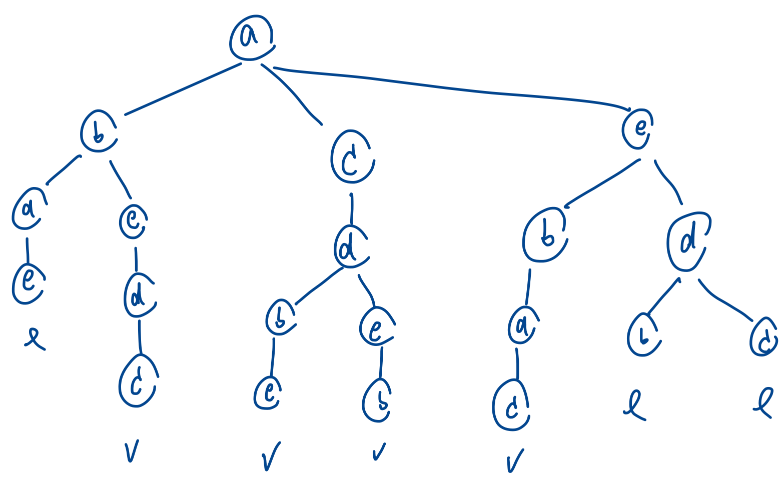
min{ m[1,1] + m[2,4] + 5x10x8 | m[1,2] + m[3,4] + 5x3x8 | m[1,3] + m[4,4] + 5x12x8 }
0 + 528 + 400 | 150 + 288 + 120 | 330 + 0 + 480
928 | 558 | 810

# Problem 6

Apply backtracking to find a Hamiltonian circuit in the following graph. Start from vertex $a$ and show your search tree, including promising and non-promising nodes.




$$\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array}\begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix} \end{array}$$



a b e d c

a c d b e

a c d e b

a e b d c

# Problem 7

Use Horspool's algorithm to search for the pattern "TCCTATTCTT" in the DNA sequence:

j=1 j=2 j=10 j5 j5 j=1 j5

"TTATAGATCTCGTATTCTTTTATAGATCTCCTATTCTT"

TCCTATTCTT
TCCTATTCTT
TCCTATTCTT
TCCTATTCTT
TCCTATTCTT
TCCTATTCTT
TCCTATTCTT ✓

TCCTATTCTT

1

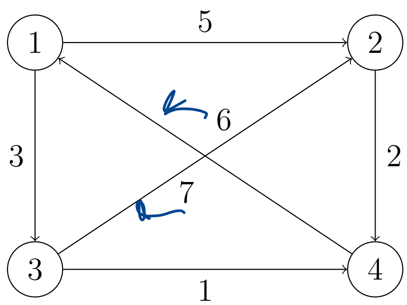
2

5

else ; 10

## Problem 8

Apply Floyd-Warshall algorithm to find the shortest path distances between all pairs of vertices in the following weighted directed graph. Show the intermediat1e matrices after each iteration.




$$A^{4} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 5 & 3 & \infty \\ \infty & 0 & \infty & 2 \\ \infty & 7 & 0 & 1 \\ 6 & \infty & \infty & 0 \end{bmatrix} \end{array}$$

$$A^{0} : \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 5 & 3 & \infty \\ \infty & 0 & \infty & 2 \\ \infty & 7 & 0 & 1 \\ 6 & \cancel{\infty}\ 11 & \cancel{\infty}\ 9 & 0 \end{bmatrix} \end{array}$$

$0 < \infty + 5 = 0$
$\infty \approx \infty + 3 = \infty$
$2 < \infty + \infty = 2$
$7 < \infty + 5 = 7$
$0 < \infty + 3 = 0$
$1 < \infty + \infty = 1$
$\infty > 6 + 5 = 11$
$\infty > 6 + 3 = 9$
$0 < 6 + \infty = 0$

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 5 & 3 & \cancel{\infty}\ 7 \\ \infty & 0 & \infty & 2 \\ \infty & 7 & 0 & 1 \\ 6 & 11 & 9 & 0 \end{bmatrix} \end{array}$$

$3 < 5 + \infty = 3$
$\infty > 5 + 2 = 7$
$\infty \approx \infty + 7 = \infty$
$1 < 7 + 2 = 1$
$6 < \infty + 11 = 6$
$9 < 11 + \infty = 9$

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 5 & 3 & \cancel{7}\ 4 \\ \infty & 0 & \infty & 2 \\ \infty & 7 & 0 & 1 \\ 6 & 11 & 9 & 0 \end{bmatrix} \end{array}$$

$7 > 3 + 1 = 4$

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 5 & 3 & 4 \\ \cancel{\infty}\ 8 & 0 & \cancel{\infty}\ 11 & 2 \\ \cancel{\infty}\ 7 & 7 & 0 & 1 \\ 6 & 11 & 9 & 0 \end{bmatrix} \end{array} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 5 & 3 & 4 \\ 8 & 0 & 11 & 2 \\ 7 & 7 & 0 & 1 \\ 6 & 11 & 9 & 0 \end{bmatrix} \end{array}$$

A

## Problem 9

How many comparisons does the brute force string matching algorithm make when searching for the pattern "124" in the text "6125212901"?

6 1 2 5 2 1 2 9 0 1

1 2 4

1 2 4

1 2 4

1 2 4

1 2 4

1 2 4

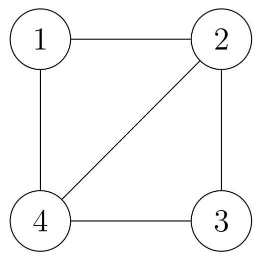
1 2 4

1 2 4

8 comparisons

# Problem 10

Consider the following graph with 4 vertices:

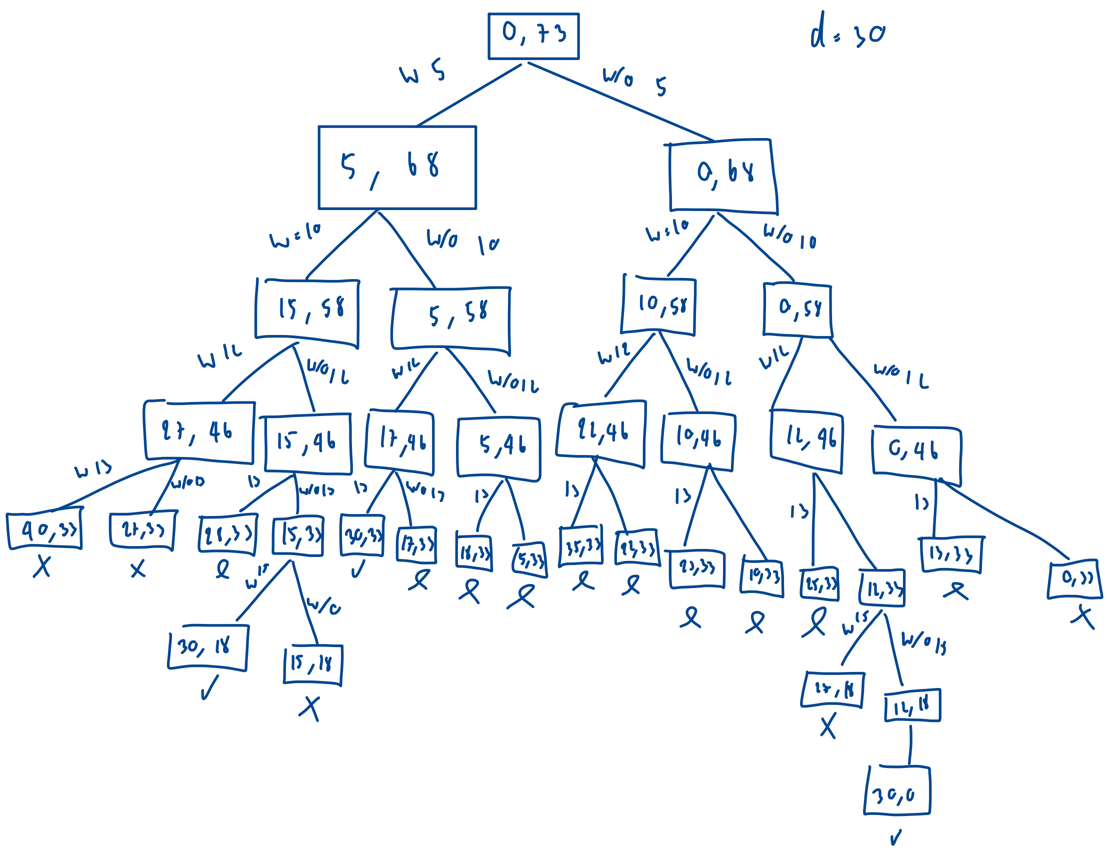



?

Apply a graph coloring algorithm to find at least one valid coloring for this graph using the 3 number of colors. Explain your approach and show your work.

## Problem 11

Apply backtracking to solve the following instance of the subset sum problem: $A = 5, 10, 12, 13, 15, 18$ and target sum $d = 30$. Find at least one subset that sums to exactly 30. Show your state space tree.



## Problem 12

For each of the following statements, indicate whether it is True or False. No need to justify your answer..

(a) Every problem in NP is also in P.

(a) **True** (b) **False**.

(b) Taylor Swift was born in 1989.

(a) **True** (b) **False**.

(c) Branch-and-bound algorithms always produce optimal solutions faster than brute-force approaches.

(a) **True** (b) **False**. ?

(d) If a problem is NP-complete, then there exists a polynomial-time algorithm to verify a solution, but no known polynomial-time algorithm to find a solution.

(a) **True** (b) **False**. ?

(e) The Floyd-Warshall algorithm can handle negative edge weights but not negative cycles.

(a) **True** (b) **False**. ?