DES227: Algorithms Design
Final Mock Exam

curated by The Peanuts

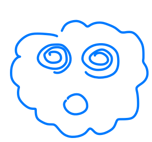
Name. . . Nonprawich I. . . ID. . . 6622772422 . . . Section. . . 2 . . . Seat No. . . . .

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.*~~



To this subject

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.

$$xy = x \cdot y = x_1 y_1 \cdot 2^n + \left[(x_1+x_0)(y_1+y_0) - x_1 y_1 - x_0 y_0\right] \cdot 2^{\frac{n}{2}} + x_0 y_0$$

n: number of bits

สมมติ $x = 1101_2$, $y = 1010_2$

$x_1 = 11$ $x_0 = 01$ $y_1 = 10$ $y_0 = 10$

$A = x_1 y_1$

$B = x_0 y_0$

$C = (x_1+x_0)(y_1+y_0)$

Simplifies

$$xy = A \cdot 2^n + (C - A - B) \cdot 2^{\frac{n}{2}} + B$$

$0100_2 \times 0001_2$

Check: $4 \times 1 = 4 = 0100_2$

$x = 0100_2$, $y = 0001_2$

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

$$xy = A \cdot 2^n + (C - A - B) \cdot 2^{\frac{n}{2}} + B$$

$A = x_1 y_1 = (01)_2 \times (00)_2 = (00)_2$

$B = x_0 y_0 = (00)_2 \times (01)_2 = (00)_2$

$C = (x_1+x_0)(y_1+y_0) = (01+00)_2 \times (00+01)_2 = (01)_2$

Shift left four times

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

$= 01 \cdot 2^2$

$= 100_2$

$= 4$ #

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.

$[5,2,8,1,9,3,7,4]$

left half $[5,2,8,1]$ → left $[5,2]$ → Sort: $[2,5]$ (1 inversion)
→ right $[8,1]$ → Sort: $[1,8]$ (1 inversion)
→ Merge $[2,5]$ & $[1,8]$

right half $[9,3,7,4]$ → left $[9,3]$ → Sort: $[3,9]$ (1 inversion)
→ right $[7,4]$ → Sort: $[4,7]$ (1 inversion)
→ Merge $[3,9]$ & $[4,7]$

$2 > 1 \rightarrow (2,1)$ ∴ 1 inversion
$2 < 8 \rightarrow$ no inversion
$5 < 8 \rightarrow$ no inversion

$3 < 4 \rightarrow$ no inversion
$9 > 4 \rightarrow (9,4)$ ∴ 1 inversion
$3 < 7 \rightarrow$ no inversion
$9 > 7 \rightarrow (9,7)$ ∴ 1 inversion

∴ 3 inversions

Merge $[1,2,5,8]$ & $[3,4,7,9]$

⋮
$5 > 3 \rightarrow (5,3)$ ∴ 1 inversion
$5 > 4 \rightarrow (5,4)$ ∴ 1 inversion

$8 > 7 \rightarrow (8,7)$ ∴ 1 inversion

∴ 3 inversions

Merged: $[1,2,3,4,5,7,8,9]$

∴ Total inversions $= 4 + 3 + 3 = 10$ inversions #

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.

	\$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	$\boxed{1}$	2	3	2	3
$d_3 = 4$	0	1	2	1	1	2	2	$\boxed{2}$
$d_4 = 5$	0	1	2	1	1	1	2	2

Starting here

∴ Uses only two coins $\{d_2, d_3\}$ #

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.

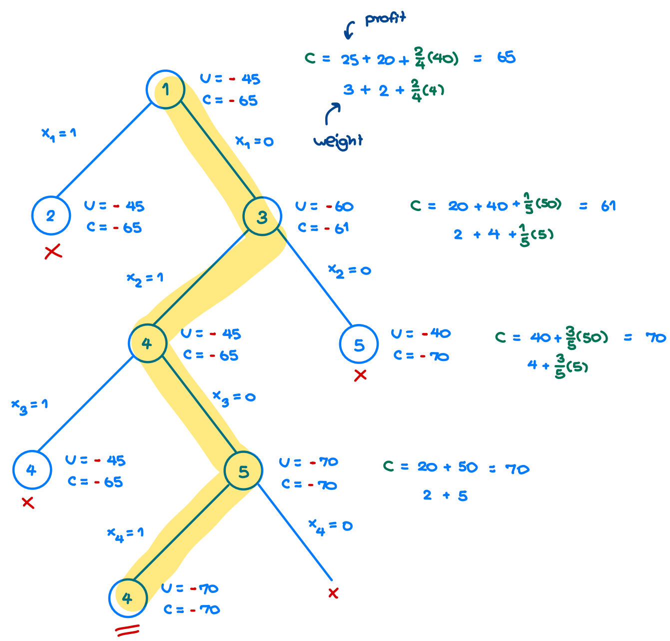
			0	1	2	3	4	5	6	7
P	w	0	0	0	0	0	0	0	0	0
25	3	1	0	0	0	25	25	25	25	25
20	2	2	0	0	20	25	25	45	45	45
40	4	3	0	0	20	25	40	45	60	65
50	5	4	0	0	20	25	40	50	60	70

$$\left\{\begin{matrix} x_1, & x_2, & x_3, & x_4 \\ 0 & 1 & 0 & 1 \end{matrix}\right\}$$ #

(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.

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

$W = 7$



$$\begin{Bmatrix} x_1, & x_2, & x_3, & x_4 \\ 0 & 1 & 0 & 1 \end{Bmatrix} \#$$

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 k=2
2		0	360	528
3			0	288
4				0

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

$\underset{5\times10}{A_1} \cdot \underset{10\times3}{A_2} \cdot \underset{3\times12}{A_3} \cdot \underset{12\times8}{A_4}$

m[1,1]

$\underset{5\times10}{A_1}$ (not multiply with anything)

m[1,2]

$\underset{5\times10}{A_1} \cdot \underset{10\times3}{A_2}$ = 150 dot products

m[2,3]

$\underset{10\times3}{A_2} \cdot \underset{3\times12}{A_3}$ = 360 dot products

m[3,4]

$\underset{3\times12}{A_3} \cdot \underset{12\times8}{A_4}$ = 288 dot products

m[1,3] two possibilities

$\left(\underset{5\times10}{A_1} \cdot \underset{10\times3}{A_2}\right) \underset{3\times12}{A_3}$

$m[1,2] + m[3,3] + 5\cdot3\cdot12$ = 150 + 0 + 180 = 330 dot products ♛ (min)

$\underset{5\times10}{A_1} \cdot \left(\underset{10\times3}{A_2} \cdot \underset{3\times12}{A_3}\right)$

$m[1,1] + m[2,3] + 5\cdot10\cdot12$ = 0 + 360 + 600 = 960 dot products

m[2,4] two possibilities

$\left(\underset{10\times3}{A_2} \cdot \underset{3\times12}{A_3}\right) \underset{12\times8}{A_4}$

$m[2,3] + m[4,4] + 10\cdot12\cdot8$ = 360 + 0 + 960 = 1,320 dot products

$\underset{10\times3}{A_2} \cdot \left(\underset{3\times12}{A_3} \cdot \underset{12\times8}{A_4}\right)$

$m[2,2] + m[3,4] + 10\cdot3\cdot8$ = 0 + 288 + 240 = 528 dot products ♛ (min)

ลองใช้สูตรบ้างนะ !

m[1,4]

$\underset{5\times10}{A_1} \cdot \underset{10\times3}{A_2} \cdot \underset{3\times12}{A_3} \cdot \underset{12\times8}{A_4}$

$d_0 \quad d_1 \quad d_2 \quad d_3 \quad d_4$

$$m[1,4] = \min \left\{ \begin{array}{c|c|c} m[1,1] + m[2,4] + 5\cdot10\cdot8, & m[1,2] + m[3,4] + 5\cdot3\cdot8, & m[1,3] + m[4,4] + 5\cdot12\cdot8, \\ 0 + 528 + 400 & 150 + 288 + 120 & 330 + 0 + 480 \\ 928 & 558 \text{ ♛ (min)} & 810 \end{array} \right\}$$

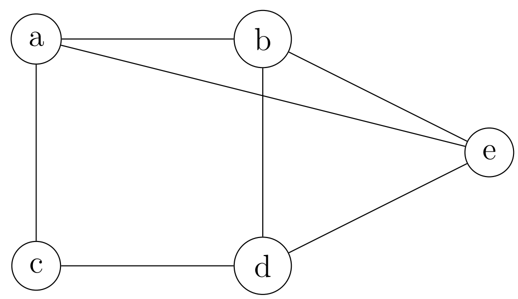
$\underset{5\times10}{A_1} \cdot \left(\underset{10\times3}{A_2} \cdot \underset{3\times12}{A_3} \cdot \underset{12\times8}{A_4}\right)$ $\left(\underset{5\times10}{A_1} \cdot \underset{10\times3}{A_2}\right)\left(\underset{3\times12}{A_3} \cdot \underset{12\times8}{A_4}\right)$ $\left(\underset{5\times10}{A_1} \cdot \underset{10\times3}{A_2} \cdot \underset{3\times12}{A_3}\right) \cdot \underset{12\times8}{A_4}$

∴ Optimal parenthesization is $\left(\underset{5\times10}{A_1} \cdot \underset{10\times3}{A_2}\right)\left(\underset{3\times12}{A_3} \cdot \underset{12\times8}{A_4}\right)$ #

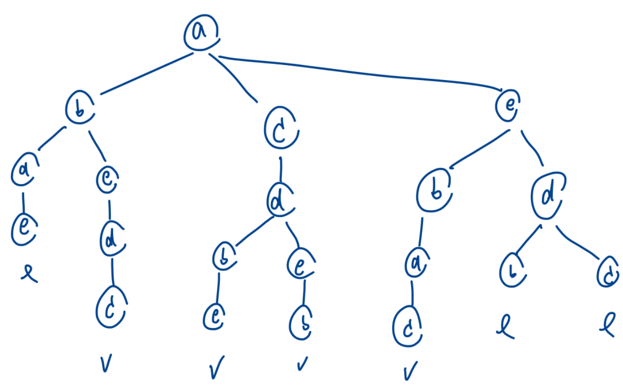
Cham's

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.



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



a b e d c

a c d b e

a c d e b

a e b d c

~~a~~

Problem 7

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

"TTATAGATCTCGTATTCTTTTATAGATCTCCTATTCTT"

```
TTATAGATCTCGTATTCTTTTATAGATCTCCTATTCTT
TCCTATTCTT
 TCCTATTCTT
   TCCTATTCTT
    TCCTATTCTT
         TCCTATTCTT
          TCCTATTCTT
           TCCTATTCTT
            TCCTATTCTT
                 TCCTATTCTT
                  TCCTATTCTT
                    TCCTATTCTT
                      TCCTATTCTT
                       TCCTATTCTT
                            TCCTATTCTT
```

```
0 1 2 3 4 5 6 7 8 9
T C C T A T T C T T ; length = 10
```

Letter	T	A	C	G
Value	1	5	2	10

default

value = length - index - 1

T = 10 - 8 - 1 = 1

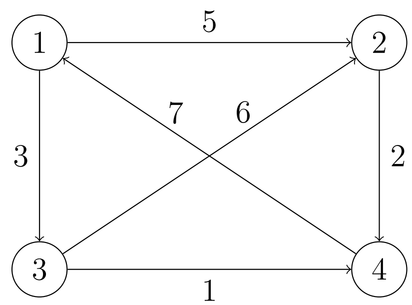
C = 10 - 7 - 1 = 2

A = 10 - 4 - 1 = 5

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.

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



$$A^1 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 5 & 3 & \infty \\ 2 & \infty & 0 & \infty & 2 \\ 3 & \infty & 6 & 0 & 1 \\ 4 & 7 & 12 & 10 & 0 \end{array}$$

$A^0[2,3] \quad A^0[2,1] + A^0[1,3]$

$\infty \approx \infty + 3$

$A^0[2,4] \quad A^0[2,1] + A^0[1,4]$

$2 < \infty + \infty$

$A^0[3,2] \quad A^0[3,1] + A^0[1,2]$

$6 < \infty + 5$

$A^0[3,4] \quad A^0[3,1] + A^0[1,4]$

$1 < \infty + \infty$

$A^0[4,2] \quad A^0[4,1] + A^0[1,2]$

$\infty > 7 + 5$

$A^0[4,3] \quad A^0[4,1] + A^0[1,3]$

$\infty > 7 + 3$

$$A^2 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 5 & 3 & 7 \\ 2 & \infty & 0 & \infty & 2 \\ 3 & \infty & 6 & 0 & 1 \\ 4 & 7 & 12 & 10 & 0 \end{array}$$

$A^1[1,3] \quad A^1[1,2] + A^1[2,3]$

$3 < 5 + \infty$

$A^1[1,4] \quad A^1[1,2] + A^1[2,4]$

$\infty > 5 + 2$

$A^1[3,1] \quad A^1[3,2] + A^1[2,1]$

$\infty \approx 6 + \infty$

$A^1[3,4] \quad A^1[3,2] + A^1[2,4]$

$1 < 6 + 2$

$A^1[4,1] \quad A^1[4,2] + A^1[2,1]$

$7 < 12 + \infty$

$A^1[4,3] \quad A^1[4,2] + A^1[2,3]$

$10 < 12 + \infty$

$$A^3 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 5 & 3 & 4 \\ 2 & \infty & 0 & \infty & 2 \\ 3 & \infty & 6 & 0 & 1 \\ 4 & 7 & 12 & 10 & 0 \end{array}$$

$A^2[1,2] \quad A^2[1,3] + A^2[3,2]$

$5 < 3 + 6$

$A^2[1,4] \quad A^2[1,3] + A^2[3,4]$

$7 > 3 + 1$

$A^2[2,1] \quad A^2[2,3] + A^2[3,1]$

$\infty \approx \infty + \infty$

$A^2[2,4] \quad A^2[2,3] + A^2[3,4]$

$2 < \infty + 1$

$A^2[4,1] \quad A^2[4,3] + A^2[3,1]$

$7 < 10 + \infty$

$A^2[4,2] \quad A^2[4,3] + A^2[3,2]$

$12 < 10 + 6$

$$A^4 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 5 & 3 & 4 \\ 2 & 9 & 0 & 12 & 2 \\ 3 & 8 & 6 & 0 & 1 \\ 4 & 7 & 12 & 10 & 0 \end{array} \quad \#$$

$A^3[1,2] \quad A^3[1,4] + A^3[4,2]$

$5 < 4 + 12$

$A^3[1,3] \quad A^3[1,4] + A^3[4,3]$

$3 < 4 + 10$

$A^3[2,1] \quad A^3[2,4] + A^3[4,1]$

$\infty > 2 + 7$

$A^3[2,3] \quad A^3[2,4] + A^3[4,3]$

$\infty > 2 + 10$

$A^3[3,1] \quad A^3[3,4] + A^3[4,1]$

$\infty > 1 + 7$

$A^3[3,2] \quad A^3[3,4] + A^3[4,2]$

$6 < 1 + 12$

Problem 9

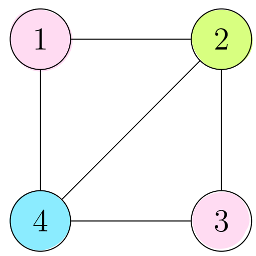
How many comparisons does the brute force string matching algorithm make when searching for the pattern “124” in the text “6125212901”?

```
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
```

$\therefore$ 12 comparisons #

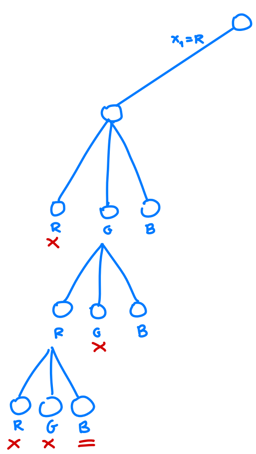
Problem 10

Consider the following graph with 4 vertices:



m = 3
{R, G, B}

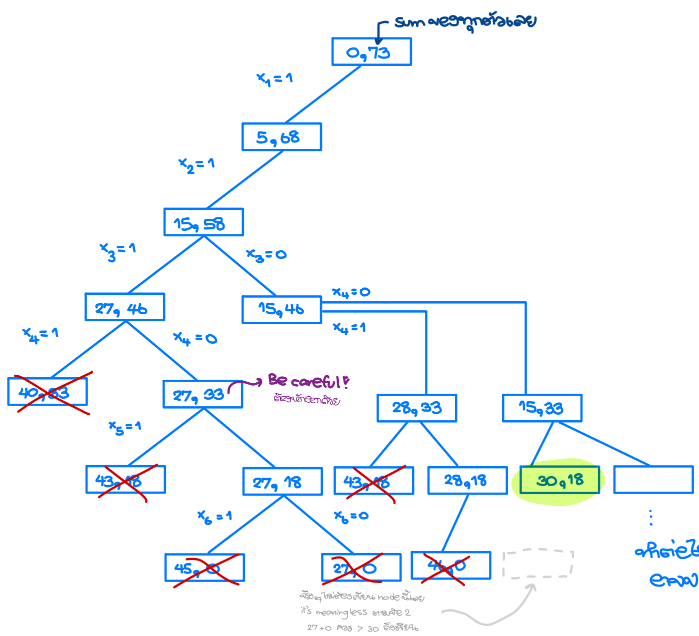
Apply a graph coloring algorithm to find at least one valid coloring for this graph using the 3 number of colors. Explain your approach and show your work.



∴ One of the solutions is RGRB #

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.



⇒ แต่ผมบอกก็รู้แล้วว่ามีอีก555

$w[1:6] = \{5, 10, 12, 13, 15, 18\}$

$n = 6 \quad m = 30$

	1	2	3	4	5	6
x	1	1	0	0	1	0

#

$x_i = 0/1$ (Value จะเป็นแค่ 0 หรือ 1 แค่นั้น)

Condition:
(Bounding f)

$$\sum_{i=1}^{k} w_i x_i + w_{k+1} \le m$$

$$\sum_{i=1}^{k} w_i x_i + \sum_{i=k+1}^{n} w_i > m$$

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. (สลับกันป่าวอ)

(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**.