Name 366-01 In-Class

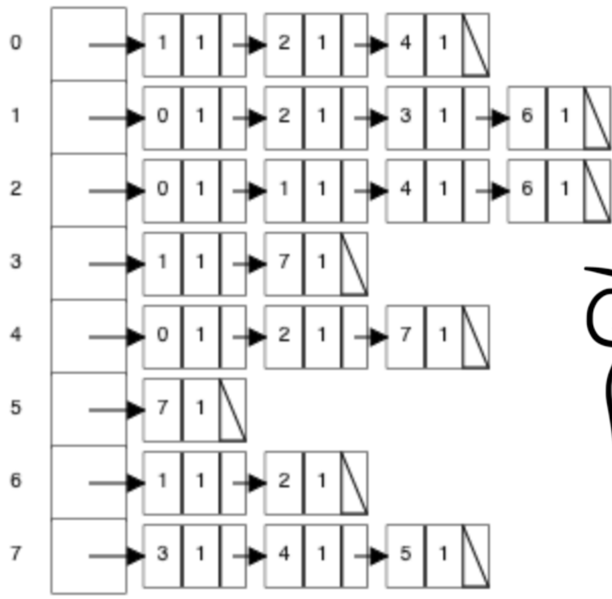
WESTERN NEW ENGLAND UNIVERSITY
COLLEGE OF ARTS AND SCIENCES

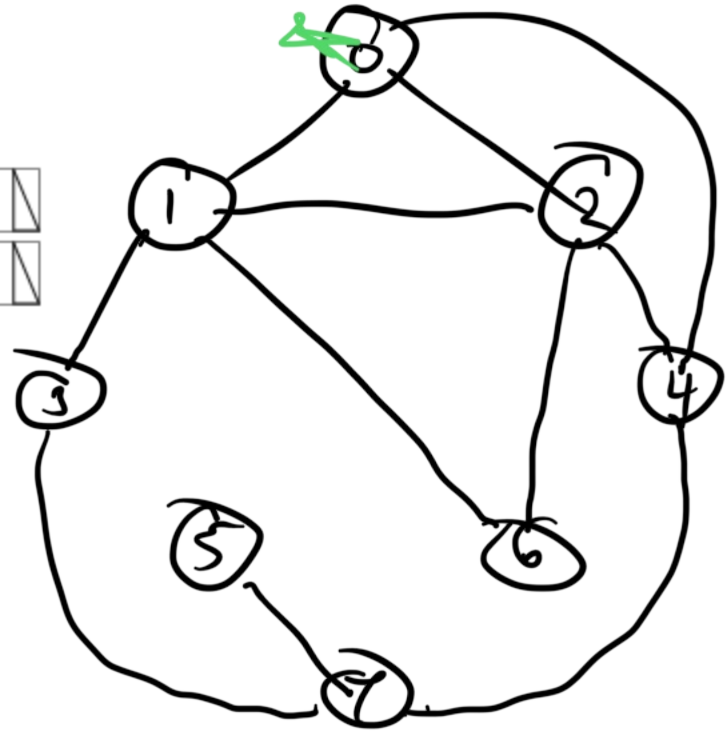
CS 366 | Design and Analysis of Algorithms | November 4, 2025

PRACTICE Exam 2

Directions: There are 50 points possible on the exam. Read the directions carefully. Write your answers in the space provided.

1.





a. (4 points) What is the visiting order when running the BFS algorithm on the above graph with **`start = 0`**?

Queue: ~~0~~, ~~1~~, ~~2~~, ~~4~~, ~~3~~, ~~6~~, ~~7~~, 5

Visited: 0, 1, 2, 4, 3, 6, 7, 5

b. (4 points) What is the visiting order when running the DFS algorithm on the above graph with **`start = 0`**?

Stack: ~~0~~, 4, 2, ~~1~~

4, 6, 3, ~~2~~

3, 6, 4, ~~7~~

~~5~~, ~~3~~

Visited: 0, 1, 2, 4, 7, 3, 5, 6

2. (4 points) Find an optimal solution to the following course interval scheduling problem.

[(A_{start}, A_{finish}), (B_{start}, B_{finish}), …]

(0, 3),	(2, 8),	(1, 4),	(3, 6),	(5, 9),	(4, 7),	(7, 11),	(8, 10),	(6, 9),	(10, 13),	(9, 12),	(11, 14)
A	B	C	D	~~E~~	~~F~~	G	H	I	J	K	L

finishing

Earliest start

Greedy = Optimal = $\boxed{\{A, D, I, K\}}$

3.

a. (8 pts.) For the function call **karatsuba(1431, 2243, 4)**, fill in each of the blanks with the appropriate integer values. Then carry out the recursion only for the recursive call to calculate A. (Each blank must contain one integer value - not an expression, not a calculation, and no variables.)

x1 = 14

x2 = 31

y1 = 22

y2 = 43

A = karatsuba(14 , 22 , 2)

B = karatsuba(31 , 43 , 2)

C = karatsuba(45 , 65 , 2)

calculate A

x1 = 1

x2 = 4

y1 = 2

y2 = 2

A = 2

B = 8

C = 20

D = 10

return 308

$(C-A)-B$

$$\begin{array}{r} 200 \\ +100 \\ +\ 8 \\ \hline 308 \end{array}$$

b. (3 pts.) Write a recurrence for the running time of the **karatsuba** algorithm and solve the recurrence using the Master Theorem (show your work). The Master Theorem is on the last page of this exam.
HINT: There are "hidden" costs: **adding** or **subtracting** two n-digit integers takes time $\theta(n)$.

$$T(n) = aT\left(\frac{n}{b}\right) + n^d$$

a = #sp b = size sp d = work done per call

$$T(n) = 3T(n/2) + n^1$$

$\log_2 3$ vs 1

$$\Theta(n^{\log_2 3})$$

c. (2 pts.) Which is more efficient: the **karatsuba** algorithm or standard long multiplication (which takes time $\theta(n^2)$)? Explain your answer.

Karatsuba is quicker

$O(n^{\log_2 3}) \approx O(n^{1.5})$ grows slower than $O(n^2)$

d. (1 pt.) Which algorithm design technique(s) (backtracking, divide-and-conquer, greedy) does the **karatsuba** algorithm use?

Divide and Conquer

4.

a. (8 pts.) Below is the table output from a trace through a portion of the algorithm **dijkstra**, with **start = 0** for an unknown weighted graph. List all the shortest paths from the start to every other node in the graph.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
key	0	10	6	8	8	16	11	14
in heap	F	F	F	F	F	F	F	F
predecessor	0	3	0	0	0	3	4	3

0: 0
1: 0, 3, 1
2: 0, 2
3: 0, 3
4: 0, 4
5: 0, 3, 5
6: 0, 4, 6
7: 0, 3, 7

b. (2 pt.) If we re-ran the algorithm dijkstra on the same graph with **start = 3**, what would the final key value (cost) be for node 7 when it is removed from the heap?
HINT: Recall that all sub-paths of a shortest path are also shortest paths

3-7

$0 \to 3 = 8$

$0 \to 7 = 14$

$3 \to 7 = 14 - 8 = 6$

$\boxed{\text{key of 6 for 7}}$

c. (2 pt.) State the running time of the algorithm **dijkstra** (which uses data structures heap and adjacency list) as a function (big Oh) of V (the number of vertices) and E (the number of edges).

$O(V+E) \cdot \log V$

$O((V+E)\log V)$

d. (2 pts.) Which algorithm design technique(s) (backtracking, divide-and-conquer, greedy) does the **dijkstra** algorithm use? **Explain** your answer (what characteristics of the algorithm indicate that particular design technique is used).

Greedy: 1) greedy choice property - visited node with lowest known cost
2) sub-optimal structure

5.

a. (2 pt.) State the running time of the algorithm **`rn_queens`** as a function (big Oh) of n (side length of board). **HINT**: Write the recurrence and solve via unrolling

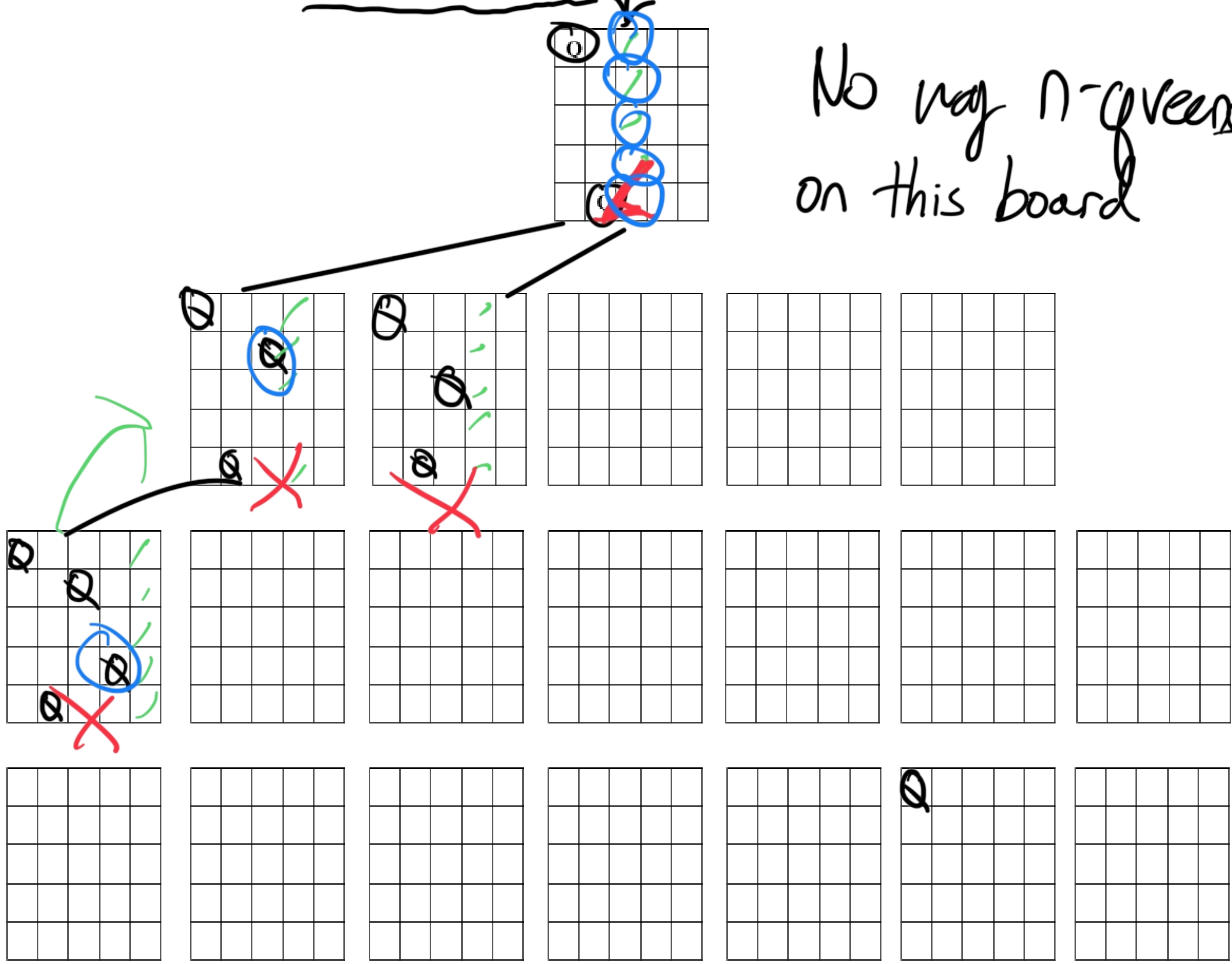
$T(n) = n\,T(n-1)$

$n \cdot n-1 \cdot n-2 \cdot n-3 \ldots$

$O(n!)$

b. (8 points) Determine if it is possible to place 3-more queens on the following n-queen problem. Only diagram boards where position_ok is true or you may run out of space.

No way n-queens on this board



BONUS (up to 4 pts.): Determine if the heap below violates the min-Heap property. Explain why or why not.

key	4	13	11	15	12	14	~~19~~	~~21~~	~~17~~	~~16~~	~~23~~
0-index	0	1	2	3	4	5	6	7	8	9	10
1-index	1	2	3	4	5	6	7	8	9	10	11

left = 2i

heapSize = 6

right = 2i + 1

node ≤ children

nodes 13, 12 violate min-Heap

$12 \ngeq 13$

The Master Theorem Suppose $T(n) = a\,T(\frac{n}{b}) + f(n)$ where $a \geq 1$ and $b > 1$.

Case 1: If $f(n) = O(n^{\log_b a - \varepsilon})$ for some constant $\varepsilon > 0$, then

$$T(n) = \Theta(n^{\log_b a})$$

Case 2: If $f(n) = \Theta(n^{\log_b a})$, then

$$T(n) = \Theta(n^{\log_b a} \log n)$$

Case 3: If $f(n) = \Omega(n^{\log_b a + \varepsilon})$ for some constant $\varepsilon > 0$ and

$a \cdot f(\frac{n}{b}) \leq c\,f(n)$ for some constant $c > 1$ and sufficiently large n, then

$$T(n) = \Theta(f(n))$$