hw11

Graded

Student

Giancarlos Marte

Total Points

26 / 40 pts

Question 1

big-O

6 / 6 pts

✔ **– 0 pts** Correct

Question 2

addition

7 / 12 pts

✔ **– 5 pts** Did not give an algorithm for problems

This is not an algorithm, this is more of an explanation, the time complexity is correct though, so just make sure you are not using examples to justify

Question 3

crossword

9 / 11 pts

✔ **– 2 pts** run time incorrect: should be some constant like $|\Sigma|^{5\times 5} = 26^{25}$

Question 4

weighted path

3 / 10 pts

✔ **– 4 pts** algorithm doesnt solve the problem, e.g., it doesnt check all possible paths or consider weights of those paths

✔ **– 3 pts** algorithm doesnt use dynamic programming

Question 5

readme

1 / 1 pt

✔ **– 0 pts** Correct

Question assigned to the following page: 1

1) Big-O drills

Big-O notation definition from lecture:

Let f and g be functions f, g: $N \rightarrow R^{+}$. Say that f(n) = O(g(n)) if positive integers c and n_0 exist such that for every integer n $\geq n_0$,

f(n) $\leq$ cg(n)

When f(n) = O(g(n)), we say that g(n) is an upper bound for f(n)...

problem	Big-O drill	true or false
1.1)	1000 = O(n^3)	true
1.2)	1000n^3 = O(n^3)	true
1.3)	n^3 = O(n^6)	true
1.4)	$1000^n = 2^{O(n)}$	true
1.5)	1000 log n = O(log n)	true
1.6)	n log n = O(log n)	false

Question assigned to the following page: 2

2) Addition algorithm

2.1)

Example numbers:

10, 11

n = 4

Algorithm:

1. separate each digit of both numbers in binary form
    a. 10 = 10, 00
    b. 11 = 10, 01
2. go through each of those digits, convert them to decimal and add them to the next
    a. (2+0+0+0)
    b. (2+0+0+1)
3. add up both totals
    a. 2 + 3 = 5

Lines 1 and 2 make it 2^n. Line 3 is just one step.

2.2)

Example numbers:

111, 101

n = 6

Algorithm:

1. setup binary numbers into 3 rows by (n/2 + 1) columns matrix
    a.

carry (row 1)	0->1	0-> 1	0 -> 1	0
row 2	0	1	1	1
row 3	0	1	0	1
results (row 4)	1	1	0	0

2. go through each column and add digits in rows 1 through 3. O(n)
    a. 1+1=0, carry a 1 and put it in row 1 of the next column
3. place the results of each column computation in row 4

Total = O(n) because you are going through each column, which is dependent on n input. The other computations are irrelevant because they are constants. O(n) is in P, which means the elementary school addition algorithm is as well.

2.3)

For unary numbers, the inputs must have the same digits. You would have to convert it into binary to make the algorithm work, then convert it back to unary. The conversions would make it exponentially slower.

Question assigned to the following page: 3

3) Crossword puzzle algorithm

3.1)

Go through each column and row, checking each letter in the alphabet.

$(5 * 5)^{26}$ steps

3.2)

The puzzle has at most 5 rows, 5 columns and each input can only be 1 of the 26 letters. Therefore, the total time complexity would be constant.

$(5 * 5)^{26}$ steps are still constant, therefore the puzzle is in P.

Question assigned to the following page: 4

4) Paths with distance

$PATH_{wgt} = \{\langle G, s, t, k\rangle | G \text{ is a graph with directed path from } s \text{ to } t \text{ with distance} \leq k\}$

step	statement	justification
1	P is the class of languages that are decidable in polynomial time…	definition of P from class lecture
2	Proof: a polynomial time algorithm M for $PATH_{wgt}$ operates as follows. M = "On input <G, s, t, k>, where G is a directed graph with nodes s and t: 1. create a weight accumulator x and set it to 0 2. place a mark on node s 3. repeat the following until no addition nodes are marked and x $\leq k$: a. scan all edges of G. If an edge (u, v) is found going from a marked node u to an unmarked node v, mark node v and add the weight value wgt to the accumulator x. 4. if t is marked and x $\leq k$, accept. Otherwise reject."	Creating an algorithm for $PATH_{wgt}$
3	# steps (worst case) n = # nodes: Line 1: 2 steps Line 2: 1 step Lines 3 and 3a: loop ● line 3: max # steps = max # edges = O(n^2) ● line 3a: loop runs at most n times ● total: O(n^3) Line 4: 1 step Total = O(n^3)	Showing time complexity for $PATH_{wgt}$
4	O(n^3) is in P, therefore $PATH_{wgt}$ is in P	1 and 3

⬛

Question assigned to the following page: 5

README
other students: none
books/websites used: class lecture slides
time spent: 4 hours