## CSCI 570 - Spring 2022 - HW2

## Due January 26th

## 1 Graded Problems

1. What is the tight upper bound to the worst-case runtime performance of the procedure below?

```
c=0

i=n

while i>1 do

for j=1 to i do

c=c+1

end for

i=\mathrm{floor}(i/2)

end while

return c
```

- 2. Arrange these functions under the O notation using only = (equivalent) or  $\subset$  (strict subset of):
  - (a)  $2^{\log n}$
  - (b)  $2^{3n}$
  - (c)  $n^{n \log n}$
  - (d)  $\log n$
  - (e)  $n \log (n^2)$
  - (f)  $n^{n^2}$
  - (g)  $\log(\log(n^n))$

E.g. for the function  $n, n + 1, n^2$ , the answer should be

$$O(n+1) = O(n) \subset O(n^2).$$

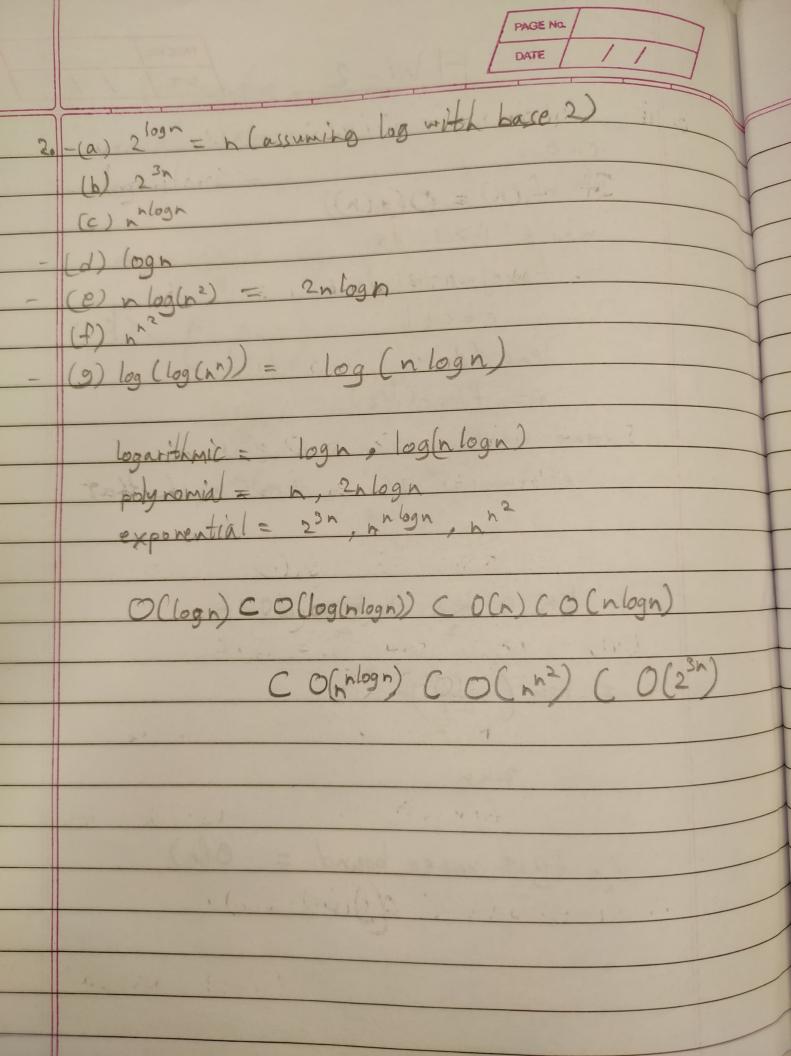
- 3. Given functions  $f_1, f_2, g_1, g_2$  such that  $f_1(n) = O(g_1(n))$  and  $f_2(n) = O(g_2(n))$ . For each of the following statements, decide whether you think it is true or false and give a proof or counterexample.
  - (a)  $f_1(n) \cdot f_2(n) = O(g_1(n) \cdot g_2(n))$
  - (b)  $f_1(n) + f_2(n) = O(\max(g_1(n), g_2(n)))$
  - (c)  $f_1(n)^2 = O(g_1(n)^2)$
  - (d)  $\log_2 f_1(n) = O(\log_2 g_1(n))$
- 4. Given an undirected graph G with n nodes and m edges, design an O(m+n) algorithm to detect whether G contains a cycle. Your algorithm should output a cycle if G contains one.

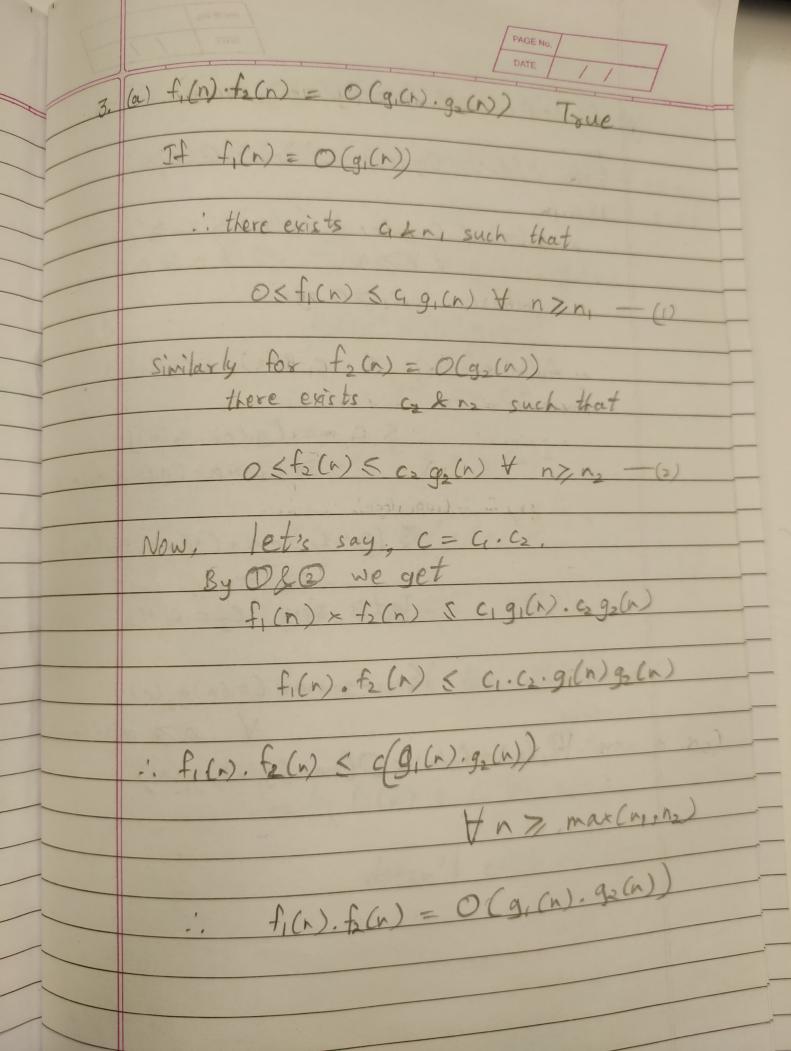
## 2 Practice Problems

- 1. Solve Kleinberg and Tardos, Chapter 2, Exercise 6.
- 2. Solve Kleinberg and Tardos, Chapter 3, Exercise 6.

PAGE No.

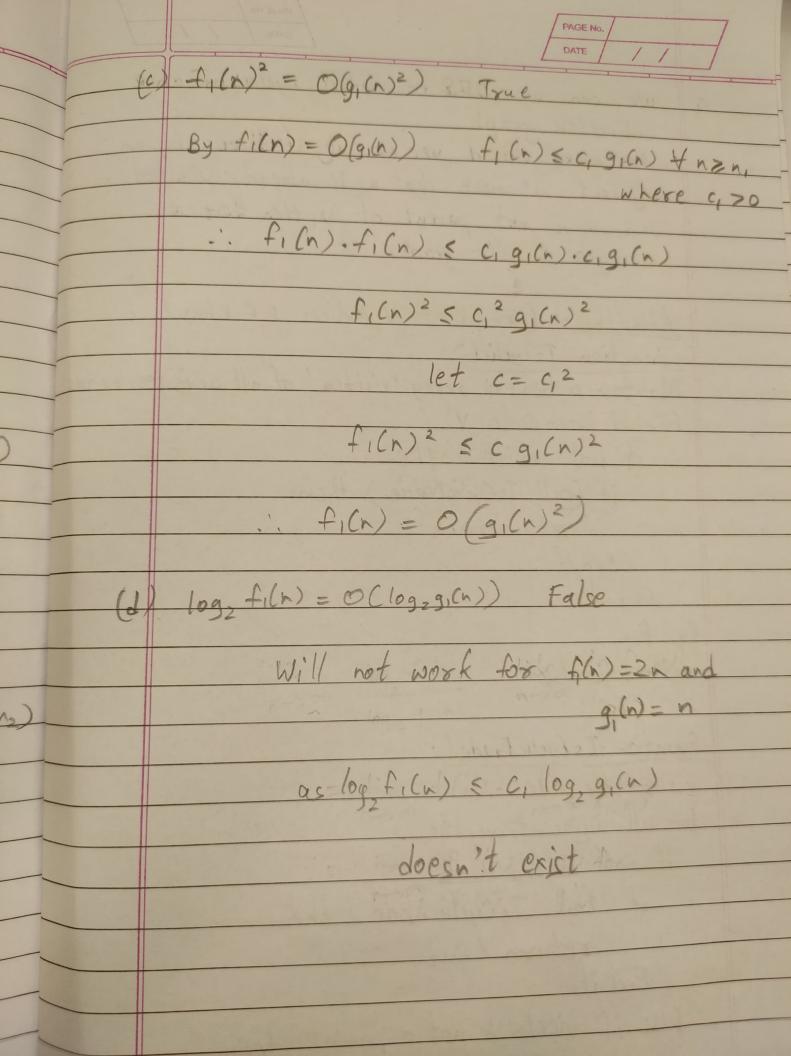
DATE HW-2 while i'> 1 do for j=1 toi do c=c+1end for 1= floor (1/2) end while return Kn + kn + kn + -= kn ( -1-1) = 2 KN i tight upper bound = O(n)





	PAGE No.
	DATE / /
	6 fi(n)+fs(n) = 0 (max(gi(n); g2(n))) True
	By $f(n) = O(g_1(n)) & f_2(n) = O(g_2(n))$
	f, (n) < c, g, (n) & f, (n) < c, g, (n)
	$\forall n \geq n$ $\forall n \geq n$
	$f_1(n) < c_1 g_1(n) & f_2(n) < c_2 g_2(n)$ $\forall n > n_1$ $\forall n > n_2$ where $c_1, c_2 > 0$
	The second secon
	: $f_1(n) + f_2(n) < c_1 g_1(n) + c_2 g_2(n)$
	S C, max (g,(n), g2(n)).
100	+ C2 max (gi(n), g2(n))
	5 (C1+C2) max (g(n), g2 (n))
	let's say $C_3 = C_1 + C_2$
	( C3 max (g(Cn),g2(n))
	Hinz max Consta
	:. f(Cn) + f2(n) = O(max(g(cn), g(n))
(-6	MARKET AND THE STATE OF THE STA
	The state of the s
	(A)

-



We can use DFS to detect cycles in an undirected graph.

for every visited vertex 'v', if there is an adjacent 'u' such that u is already visited and u is not pasent of v, then there is a cycle in the graph. Algorithm: (Graph of V vertices & F edges) Maintain a boolean array 'visited' of all unvisited nodes For i from 0 to V if not visited then if (call Is Cyclic Node) then yetron true Endif Endit End For return forlse Function Is Eyclic Node () visit the node for all adjacent vertices if not visited then if (call Is Cyclic Node) then return true Else if visited & not a parent then seturn true End if return false End for