46/100

Computer Science 303 **Exam 1** **Paul Cao**

2/21/2014, Friday in class This is a close-book close-notes exam.

Your name Junjun

TIME LIMIT: 50 MIN

Please read the following instructions carefully.

You have 50 minutes to work on this exam, which has 3 problems. The total points of this exam are 100. **You may use a 3x5 card with notes written on one side during this exam.** Other reference sources, such as textbook, in class notes, homework solutions, online resources, and discussions with other students are not allowed. You may use blank sheets of paper for scratch-work, but please put your answers and the work you want to be graded on the exam. You may get partial credits if you show your work. **You must sign the honor code in the provided space under the last problem. The honor code is "I affirm that I have adhered to the Ashland University honor code on this exam".**

For your reference, a list of useful mathematical formula is attached at the end of the exam.

Good luck!

1. Basic Efficiency analysis (37 points)

(a). (7 points)
Prove the following property using the definition of asymptotic notations. You won't receive credits if you just demonstrate the idea without proof.

If f(n) is O(g(n)) and g(n) is O(h(n)), then f(n) is O(h(n)).

set element $e \in f(n)$?

$\therefore f(n)$ is $O(g(n))$

$\therefore e \in g(n)$

$\therefore g(n)$ is $O(h(n))$

$\therefore e \in O(h(n))$

$\therefore e \in f(n)$

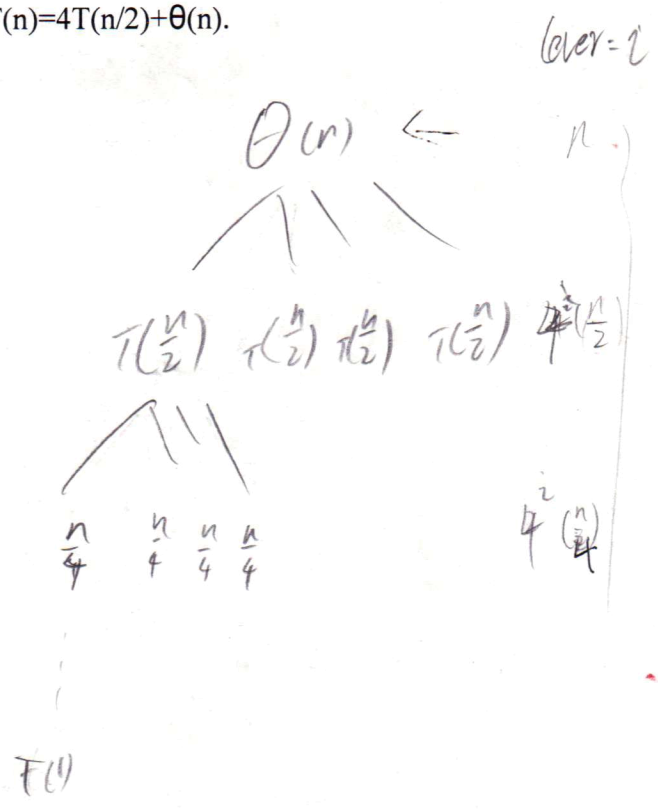
$\therefore f(n) \in O(h(n))$

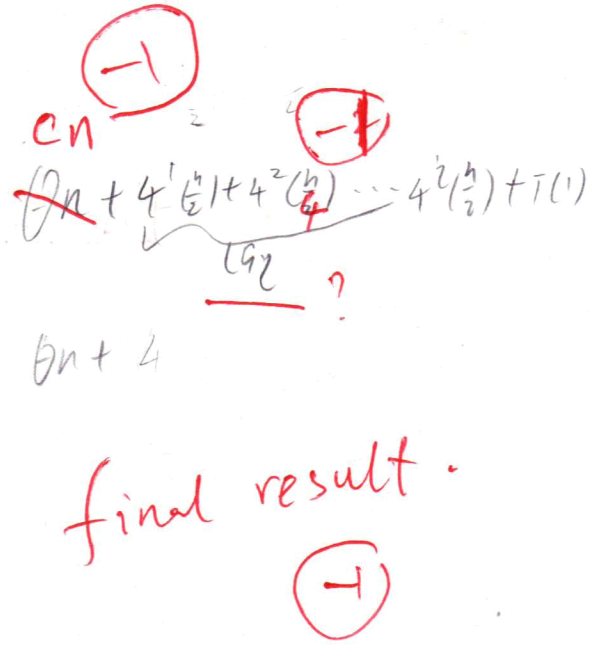
This isn't a strict proof.

-40

(b). (8 points)
Use the recursion tree method or substitution method to find the efficiency of T(n) if T(n) follows the recursion below. The answer can be in O or θ notation. You can't use the master theorem to prove this problem. You must show your steps.

T(n)=4T(n/2)+θ(n).





(c). (12 points)

Use the master theorem to find the efficiencies of the following recursions. $n^{\log_b a}$

i. $T(n)=2T(n/2)+\theta(\lg n)$

$a=2 \quad b=2 \quad f(n)=\theta(\lg n)$

RCO ? -3

ii. $T(n)=T(n/3)+\theta(n)$

$a=1 \quad b=3 \quad f(n)=\theta(n)$

$n^{\log_b(a)} = n^{\log_3} = n^0 = 1$

answer? -2

iii. $T(n)=4T(n/2)+\theta(n^2)$

$a=4 \quad b=3 \quad f(n)=\theta(n^2)$

$T(n) = \theta(n^2 \lg(n))$ ✓

(d). (10 points)
State whether the following statements are true or false. You don't have to justify your answers.

- $\log_3 n$ and $\log_2 n$ belong to the same efficiency class

true

- 2^n and 3^n belong to the same efficiency class

true X -4

- $\theta(n!)$ is worse than $\theta(2^n)$

true

- $\theta(\lg n)$ is worse than $\theta(n^{.01})$

true X -

- $\theta(\lg(n^2))$ is the same as $\theta((\lg n)^2)$

false

2. Sorting algorithms (30 points)
(a). (10 points)
Apply merge-sort to sort the list A, L, G, O, R, I, T, H, M, S. You must give intermediate steps or else no credit will be given.

A L G O R I T H M S

AGLORITHMS ? -10

(b). (10 points)
Use heapsort to sort the following array. You must show intermediate steps to receive credits. You can use either heap structure as a ~~graph~~ or an array.

20, 40, -5, -1, 2

1 2 3 4 5

40 | 20, -5, -1, 2

1 2 3 4 5

1, 2, 3

2, 4, 5

40

/ \

20 -5

/

-1 2

Sorting? -6

(c). (10 points)
For the following algorithms, write the recursion formula for their worst-case scenario and describe when it happens. Write out the worst-case efficiency. You don't have to justify your answer for the efficiency though you may receive partial credits for that.

- quicksort

$\Theta(n^2)$ ✓ when? recursive?

$\Theta(n \lg n)$ -2

- binary search

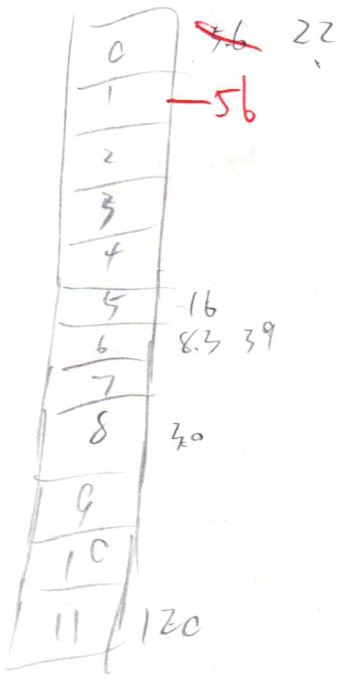
-5

3. Hashing (33 points)

(a). (10 points)

Use open hashing to hash the following integers.
The hash function is h(k)= k mod 11.

56, 16, 83, 30, 39, 22, 120



(b). (11 points)

Use closed-hashing approach to hash the following integers. The hashing function should use quadratic probing as $h(k, i)=(k \bmod 13 + 3i^2+i) \bmod 13$

7, 24, 20, 0, 73, 51

7 9 7 0 8 12

13|73 5, 65, 8

13|51 3, 39, 12

0 1 2 3 4 5 6 7 8 9 10 11 12 13

0 ✓ 7 ✓ ~~73~~ ~~24~~ 51

-7

(c). (12 points)
Using multiplication hashing function $h(k)=(k*A \bmod 2^{7}) >> (4-r)$ where m=8, r=3, w=7. Use open hashing to fill in the hashing table.

$k_1*A=1101100\ 1110001$
$k_2*A=1110011\ 1010011$
$k_3*A=0000001\ 1010000$
$k_4*A=1110001\ 1100000$
$k_5*A=1110001\ 0010011$
$k_6*A=1000001\ 0111111$

Wrong position! -6

~~Wrong~~

0 1 2 3 4 5 6 7

k_3/A k_1/A k_4/A k_2/A k_5/A k_6/A

Write and sign the honor code here.

I affirm that I have adhered to the Ashland University honor code on this exam