NATIONAL UNIVERSITY OF SINGAPORE
SCHOOL OF COMPUTING

TERM TEST FOR
Semester 2, AY2018/2019

CS3230 – DESIGN AND ANALYSIS OF ALGORITHMS

9 Mar 2019 Time Allowed: 2 hours

Instructions to Candidates:

1. This paper consists of **FOUR** questions and comprises **TWELVE (12)** printed pages, including this page.
2. Answer **ALL** questions.
3. Write ALL your answers in this examination book.
4. This is an **OPEN BOOK** examination.

Matric. Number: ____________________
Tutorial Group Number: ____________________

QUESTION	POSSIBLE	SCORE
Q1	20	
Q2	20	
Q3	30	
Q4	20	
TOTAL	**90**	

IMPORTANT NOTE:

- *You can* ***freely quote*** *standard algorithms and data structures covered in the lectures and homeworks. Explain* ***any modifications*** *you make to them.*
- *Unless otherwise specified, you are expected to* ***prove (justify)*** *your results.*

Q1. (20 points) Sorting out order of growth rates

(a) Rank the following functions in *increasing order of growth*; that is, if function $f(n)$ is *immediately* before function $g(n)$ in your list, then it should be the case that $f(n)$ is $O(g(n))$.

$g_1(n) = \sum_{i=2}^{n} \frac{n^3}{i(i-1)}$ $\quad g_2(n) = n^2 \lg\lg n$ $\quad g_3(n) = n!$ $\quad g_4(n) = n^{\lg n}$

$g_5(n) = 2^{2\cdot 2^{\lg\lg n}}$ $\quad g_6(n) = (\lg n)^n$ $\quad g_7(n) = n^3$ $\quad g_8(n) = 2^n$

To simplify notations, we write $f(n) << g(n)$ to mean $f(n) = o(g(n))$ and $f(n) \equiv g(n)$ to mean $f(n) = \Theta(g(n))$. For example, the four functions n^2, n, $(2013n^2 + n)$ and n^3 could be sorted in increasing order of growth as follows: ($n << n^2 \equiv (2013n^2 + n) << n^3$). *Proofs are not required for this problem.*

Soln: $2^{2\cdot 2^{\lg\lg n}} \ll n^2 \lg\lg n \ll n^3 \equiv \sum_{i=2}^{n} \frac{n^3}{i(i-1)} \ll n^{\lg n} \ll 2^n \ll (\lg n)^n \ll n!$

(b) Can you show that $n \lg\lg n = O(n \lg n)$ by limit method?

Soln: $\lim_{n\to\infty} \frac{n \lg\lg n}{n \lg n} = \lim_{n\to\infty} \frac{\lg\lg n}{\lg n} = \lim_{n\to\infty} \frac{\left(\frac{d \lg\lg n}{d \lg n} \cdot \frac{d \lg n}{dn}\right)}{\left(\frac{d \lg n}{dn}\right)} = \lim_{n\to\infty} \frac{\left(\frac{1}{n \lg n}\right)}{\left(\frac{1}{n}\right)} = \lim_{n\to\infty} \frac{1}{\lg n} = 0.$

Hence, $n \lg\lg n = O(n \lg n)$.

- Most people got this correct. 5 obtainable marks
- One common mistake amongst those who got it wrong was differentiating log log n incorrectly. Along those lines, a lot of people made the claim that n over log n converges. Which is completely untrue.
- 5 marks for solutions that are either as above, or explicitly stated that log n is strictly smaller than n, and thus it converges.
- 0 marks for either wrong steps or unjustified steps including jumping straight from log n over n converges.

Q1. (continued…)

(c) Can you show that $3n^2 + 2^{3\lg n}\left(1+\cos\left(\frac{n\pi}{2}\right)\right) = O(n^3)$ by definition? Please show the steps and state clearly what are c and n_0.

Soln: n_0=1, c=5.

$$3n^2 + 2^{3\lg n}\left(1+\cos\left(\frac{n\pi}{2}\right)\right) = 3n^2 + n^3\left(1+\cos\left(\frac{n\pi}{2}\right)\right) \leq 3n^2 + 2n^3 \leq 5n^3$$

- A very common mistake here was setting c to be 2 or less. When n is a multiple of 4, the above function becomes $3n^2 + 2n^2$, which is larger than $2n^3$. This would not imply what is needed to be shown by definition. For other values of c, n_0 still needs to be chosen to be sufficiently large.
- Along those lines, some people had also just said this works for any n_0 >= 0. Which is not the case.
- Additionally, some people did not include the values when explicitly asked to do so.
- 5 obtainable marks. 5 for correct solution, 3 marks for those who only wrote down either c or n_0 but not both. 0 marks for wrong values of c or n_0.

Q2. (20 points) Solving recurrence

(i) $T(n) = 2\,T(n/2) + n \lg \lg n$
(ii) $T(n) = 4\,T(n/2) + n$
(iii) $T(n) = 4\,T(n/2) + n \lg \lg n$
(iv) $T(n) = 4\,T(n/2) + n^2$
(v) $T(n) = 4\,T(n/2) + n^2 \lg \lg n$
(vi) $T(n) = 4\,T(n/2) + n^3$
(vii) $T(n) = 4\,T(n/2) + n^3 \lg \lg n$

(a) Among the above recurrences, please indicates those that can be solved by master theorem. For each recurrence that is solvable by master theorem, please indicate which case it belongs to (either case 1, 2, or 3) and state the time complexity.

Soln: The solution is as follows.

(i) $T(n) = 2\,T(n/2) + n \lg \lg n$:
NO!

(ii) $T(n) = 4\,T(n/2) + n$:
YES! Case 1, $\Theta(n^2)$.

(iii) $T(n) = 4\,T(n/2) + n \lg \lg n$:
YES! Case 1, $\Theta(n^2)$.

(iv) $T(n) = 4\,T(n/2) + n^2$:
YES! Case 2, $\Theta(n^2 \log n)$.

(v) $T(n) = 4\,T(n/2) + n^2 \lg \lg n$:
NO!

(vi) $T(n) = 4\,T(n/2) + n^3$:
YES! Case 3, $\Theta(n^3)$.

(vii) $T(n) = 4\,T(n/2) + n^3 \lg \lg n$:
YES! Case 3, $\Theta(n^3 \lg \lg n)$.

Comments:

1. 2 marks for each for those that can be solved with master theorem. 1 for case and 1 for time complexity
2. Most of you got this right. Some of you used master theorem for (i). Plz refer to lecture notes of week-2 for why MT is not applicable for this case. Similarly, for v.

Q2. (continued…)

(b) For recurrences that are not solvable by master theorem, please solve them.

Soln:

For (i), T(n) = 2 T(n/2) + n lg lg n.
We can solve it using telescoping. Dividing both sides of equation (i) by n, we have:

$\frac{T(n)}{n} = \frac{T(n/2)}{n/2} + \lg\lg n.$

By telescoping, we also have:

$\frac{T(n/2)}{n/2} = \frac{T(n/4)}{n/4} + \lg\lg(n/2).$

$\frac{T(n/4)}{n/4} = \frac{T(n/8)}{n/8} + \lg\lg(n/4).$

$\frac{T(n/8)}{n/8} = \frac{T(n/16)}{n/18} + \lg\lg(n/8).$

…

$\frac{T(2)}{2} = \frac{T(1)}{1} + \lg\lg(2).$

By summing over all equations, we have:

$\frac{T(n)}{n} = \frac{T(1)}{1} + \lg\lg n + \lg\lg(n/2) + \cdots + \lg\lg(2).$

$\frac{T(n)}{n} = \frac{T(1)}{1} + \lg((\lg n)!).$

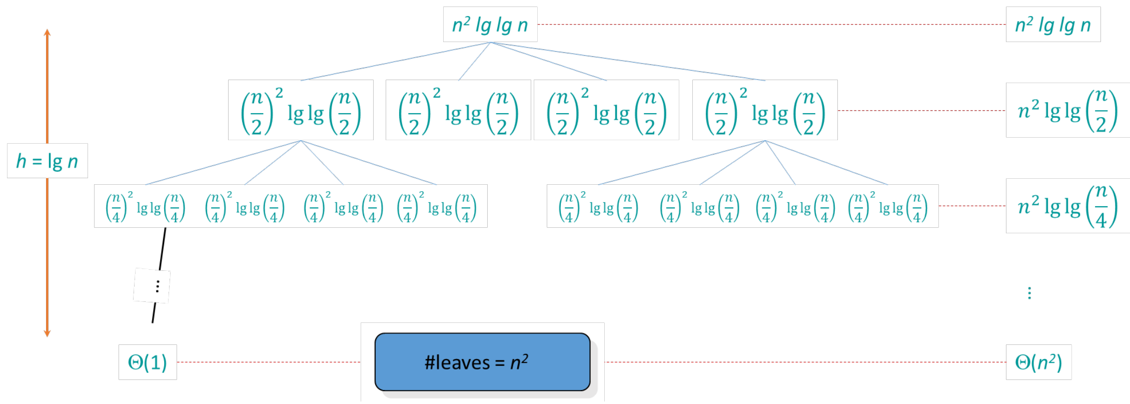
Hence, $T(n) = \Theta(n \lg n \lg\lg n)$.

Comments:

1. 5 marks each for (i) and (v)
2. Most of you who could figure out MT not applicable for this case, could expand the recurrence as shown above but could not solve the summation of lg lg i , i =1..lg n .
3. Partial marks are given for those who did the first part of expanding recurrence and 1 more mark for those who wrote the summation in correct form including the range of i.

Q2. (continued…)

For (v), $T(n) = 4\,T(n/2) + n^2 \lg\lg n$



$$T(n) = \sum_{i=0}^{\lg n-1} n^2 \lg\lg\left(\frac{n}{2^i}\right) = n^2 \sum_{i=0}^{\lg n-1} \lg(\lg n - i) = n^2 \lg((\lg n)!) = \Theta(n^2 \lg n \lg\lg n)$$

Comments:

1. This is similar to i. except n is squared in f(n). Similar marking is followed as in (i)
2. Most of you who did not do this could not recognize the similarity between (i) and (v).

Q3. (30 points) Divide-and-conquer

You are given an unsorted integer array A[1..n] of length n. Assume all integers in A[1..n] are distinct. Consider an array of m indices $s[1..m]$ such that $1 < s[1] < s[2] < \cdots < s[m] < n$. We would like to return the $s[k]$-th smallest elements of A[1..n] for all $k \in \{1, 2, \ldots, m\}$.

(a) Consider the following algorithm Select(A[1..n], s[1..m]) that return D[1..m] where D[k] is the s[k]-th smallest element in A[1..n]. Can you show that the algorithm is correct?

Select(A[1..n], s[1..m])

1. If (m==0) then return null;
2. Let $k = \lfloor (1 + m)/2 \rfloor$;
3. By linear select, we obtain x = the s[k]-th smallest element in A[1..n];
4. D[k] = x
5. D[1..k-1] = Select($A[1..n], s[1..k-1]$)
6. D[k+1..m] = Select ($A[1..n]$, s[k+1..m])
7. Return D[1..m];

Soln: We can show the correctness by induction.
P(m): Select(A[1..n], s[1..m]) return correct answer.

Base case: When m=0 (i.e. s is an empty array), the algorithm return null correctly.

Inductive case: When m>0 (i.e. s is a length-m array), assume Select(A[1..n], s[1..q]) is correct for q < m.
Let $k = \lfloor (1 + m)/2 \rfloor$.
Steps 3-4 compute D[k] correctly.
Step 5 computes D[1..k-1] correctly by inductive hypothesis.
Step 6 computes D[k+1..m] correctly by inductive hypothesis.
Hence, P(m) is true.

Q3. (continued…)

(b) Can you give the time complexity of Select(A[1..n], s[1..m])?

Soln:
Let T(m) be the running time of Select(A[1..n], s[1..m]).
The recurrence is T(m) = 2 T(m/2) + cn.
By master theorem, T(m) = cnm = Θ(nm).

(c) Can you propose an efficient algorithm DIVIDE(r, A[1..n]) that computes (x, $B[1..r-1]$, $C[1..n-r]$)?

a. x is the r-th smallest element in $A[1..n]$,
b. $B[1..r-1]$ is an array that contains all elements in A smaller than x, and
c. $C[1..n-r]$ is an array that contains all elements in A bigger than x.

What is the running time of DIVIDE(r, A[1..n])?

Soln: Below algorithm runs in O(n) time.

DIVIDE(r, A[1..n])

1. Compute the r-th smallest elements x of $A[1..n]$ using linear time select;
2. For i= 1 to n,
    - if A[i]<x then insert into B, else insert into C;
3. Return (x, $B[1..r-1]$, $C[1..n-r]$);

Q3. (continued…)

(d) Refer to part (c), if y is the q-th smallest element in A[1..n], can you answer the following queries?

i. If q>r, what is the rank of y in $C[1..n-r]$? (note: y is rank j if y is the j-th smallest element in $C[1..n-r]$.)

ii. If q<r, what is the rank of y in $B[1..r-1]$?

Soln: (i) y is of rank (q-r) in $C[1..n-r]$.
(ii) y is of rank q in $B[1..r-1]$.

Q3. (continued…)

(e) Can you propose an O(n lg m)-time algorithm for Select(A[1..n], S[1..m])?

Soln: The algorithm is as follows.

Select(A[1..n], s[1..m])
1. If (m==0) then return null;
2. Let k = (1+m)/2;
3. (x, B[1..s[k]-1], C[1..n-s[k]])=DIVIDE(s[k], A[1..n]);
4. Let t[1..m-k] be an array such that t[i] = s[k+i]-s[k];
5. D[1..k-1] = Select($B[1..s[k]-1], s[1..k-1]$)
6. D[k] = x
7. D[k+1..m] = Select ($C[1..n-s[k]]$, t[1..m-k])
8. Return D[1..m];

$T(m,n) = T(m/2,n_1)+T(m/2,n_2) + \Theta(n)$.
By master theorem, this algorithm runs in O(n lg m) time.

(a) (8 marks)
- Some students do not give an inductive proof. (3 marks)
- Some students prove the correctness using invariant. It is not correct. (I give 2 marks).
- Some students give the base case, but fail to give the inductive step. (I give 2 marks)
- Some students do not give or give an incorrect inductive hypothesis. (I give 5 marks.)

(b) (5 marks)
- Some students said the time complexity is Θ(n lg n), which is not correct.
- Recurrence is correct T(n,m)=2T(n,m/2)+cn. Time complexity is wrong (3 marks)

(c) (5 marks)
- Some students use randomized select. It cannot give good worst case time. It can be $O(n^2)$. (I give 3 marks)
- Some students sort all numbers first and give an O(n lg n) time algorithm for DIVIDE. (I give 3 marks)

(d) (2 marks)
- Some students report q-1 or q-r-1.

(e) (10 marks) Don’t have the steps to calculate the t[] array. (I gives 8 marks)

Q4. (20 points) Randomized Algorithms

(a) On a rainy day in Singapore, n students enter a restaurant, each depositing an umbrella in a bin. The umbrellas get shuffled around, so that when the students come out, the receptionist hands out the umbrellas in random order. What is the expected number of students who get back their own umbrella?

Soln: Let X_j be the indicator variable for the event that person j gets their own umbrella back. $E[X_j] = \Pr[X_j = 1] = 1/n$.
Then, the expected number of people who got their own umbrella back is:

$$E\left[\sum_{j\in[n]} X_j\right] = \sum_{j\in[n]} E[X_j] = n \cdot \frac{1}{n} = 1.$$

(10 marks)

- I mostly gave either 0 or 10 depending on the final answer.
- A very few students wrote $\sum \frac{1}{n}$ but wrote something other than 1. For this they got 6 points.

Q4. (continued…)

(b) Xin Yi goes to a casino. She starts out with a net profit of 0 and plays a game with a sequence of rounds. In each round, her profit increases by 1 with probability 1/9 and decreases by 1 with probability 8/9. Each round is independent.
Give an upper-bound on the expected number of rounds in which her net profit is positive.

Soln: Define an indicator variable X_i that is 1 iff the net profit is positive

We want to compute $E[\sum X_i] = \sum E[X_i]$.

$X_i = 1$ if Xin Yi wins in more than $i/2$ of the rounds between 1 and i.

Therefore:

$$\Pr[X_i = 1] = \Pr\left[\exists S \subseteq [i], |S| > \frac{i}{2}, \text{wins in all rounds } j \in S\right]$$
$$\leq \sum_{S\subseteq[i],|S|>\frac{i}{2}} \Pr[\text{wins in all rounds } j \in S]$$
$$\leq 2^i \cdot 9^{-\frac{i}{2}}$$
$$\leq \left(\frac{2}{3}\right)^i$$

Therefore, $E[\sum X_i] \leq \sum_i \left(\frac{2}{3}\right)^i = 3$.

(10 marks)

- Full marks were given for correct reasoning followed by a small constant upper bound
- Correct reasoning but without an actual bound got 8 marks
- Almost correct reasoning got 4 to 6 marks, depending on the error
- No points were given if the answer addressed a different question.

-- End of Paper ---