HKUST – Department of Computer Science and Engineering

COMP3711: Design and Analysis of Algorithms – Fall 2017

#

Final Examination

Date: Friday, Dec. 8, 2017 Time: 08:30–11:30 Venue: LG1 Table Tennis Room

Name: ______________	Student ID: ______________
Email: ______________	Lecture L1 / L2

Instructions

- This is a closed book exam. It consists of 17 pages and 11 questions .
- Please write your name, student ID and ITSC email and circle your lecture section (L1 is M,W 9-9:20 and L2 is W, F 15:00-16:20) at the top of this page.
- For each subsequent page that you write on, please write your student ID at the top of the page in the space provided.
- Please sign the honor code statement on page 2.
- Answer all the questions within the space provided on the examination paper. You may use the back of the pages for your rough work. The last 2 pages are scrap paper and may also be used for rough work. Each question is on a separate page and most have at least one extra page for writing answers. These are provided for clarity and are not meant to imply that each question requires all of the blank pages. Many can be answered using much less space.

Questions	1	2	3	4	5	6	7	8	9	10	11	Total
Points	8	10	9	10	10	10	7	8	10	12	6	100
Score												

Student ID: ____________

As part of HKUST's introduction of an honor code, the HKUST Senate has recommended that all students be asked to sign a brief declaration printed on examination answer books that their answers are their own work, and that they are aware of the regulations relating to academic integrity. Following this, please read and sign the declaration below.

I declare that the answers submitted for this examination are my own work.

I understand that sanctions will be imposed, if I am found to have violated the University regulations governing academic integrity.

Student's Name: ______________________________

Student's Signature: ________________________

Student ID: ____________

1. **Time Complexity [8 pts]**

We have two algorithms, A and B. Let $T_A(n)$ and $T_B(n)$ denote the time complexities of algorithm A and B respectively, with respect to the input size n. Below there are 8 different cases of time complexities for each algorithm. Complete the last column of the following table with "A", "B", or "U", where:

- "A" means that for large enough n, algorithm A is always faster;
- "B" means that for large enough n, algorithm B is always faster;
- "U" means that the information provided is not enough to justify stating that, for large enough n, one algorithm is always faster than the other.

Case	$T_A(n)$	$T_B(n)$	Faster
1	$\Theta(n^{1.5})$	$\Theta(n^2/(\log n)^3)$	
2	$O(n^2)$	$\Omega(2^{\sqrt{n}})$	
3	$O(\log n)$	$\Theta(2^{\log\log n})$	
4	$\Theta(\log^3 n)$	$\Theta(\sqrt[4]{n})$	
5	$O(n^4)$	$O(n^{2.5})$	
6	$\Theta(2^{2n})$	$\Theta(\sqrt{6^n})$	
7	$\Omega(n^{3.1})$	$O(n^{2.8})$	
8	$\Theta(n^3)$	$\Theta(4^{\log_5 n})$	

Solution: 1) A, 2) A, 3) U, 4) A, 5) U, 6) B, 7) B, 8) B

Student ID: ________

2. **Divide-and-Conquer [10 pts]**

Suppose you are given an array $A[1..n]$ of sorted integers that has been circularly shifted k positions to the right. For example, $[35, 42, 54, 5, 27, 29]$ is a sorted array that has been circularly shifted $k = 3$ positions, while $[29, 35, 45, 54, 5, 27]$ has been shifted $k = 4$ positions. We can obviously find the largest element in A in $O(n)$ time. Describe an $O(\log n)$ algorithm based on the divide-and-conquer strategy to find the largest element in A.

Solution:

```
FindMax($A[1..n], s, e$)
if $s == e$ then
    return $A[s]$.
$m = \lfloor (s+e)/2 \rfloor$
if $A[m] > A[m+1]$ then
    return $A[m]$
if $A[m] > A[s]$
    return FindMax($A[1..n], m, e$);
else
    return FindMax($A[1..n], s, m$);
```

The initial call is FindMax$(A, 1, n)$. This algorithm runs in $O(\log n)$.

There is also a $O(\log k)$ algorithm.

Student ID: ________

3. **Heap [9 pts]**
   The following array stores a Min-Heap: $[5, 18, 15, 120, 20, 21]$.

   (a) Please show the array of the Min-Heap after performing Extract-min from the original Min-heap.

   (b) Please show the array of the Min-Heap after 14 is inserted into the original Min-heap.

   (c) Please show the array of the Min-Heap after performing Decrease-key to decrease 20 to 3 from the original Min-heap.

Solution: (a) $[15, 18, 21, 120, 20]$.

(b) $[5, 18, 14, 120, 20, 21, 15]$.

(c) $[3, 5, 15, 120, 18, 21]$.

Student ID: __________

4. **Dynamic programming [10 pts]**
A message containing letters from $A - Z$ is being encoded to numbers using the following mapping:
"A" $\rightarrow 1$
"B" $\rightarrow 2$
...
"Z" $\rightarrow 26$

Given an encoded message A containing n digits, design a $O(n)$ time dynamic programming algorithm to determine the total number of ways to decode it.

For example, Given encoded message "23", it could be decoded as "BC"(2, 3) or "W"(23). The number of ways decoding "23" is 2.

Solution: For $0 \leq i \leq n$, define $d[i]$ to be the total number of ways to decode the first i digits (i.e. $A[1..i]$).

Base case: $d[0] = 1, d[1] = 1$
Recursive case:
$d[i] = d[i-1]$ if $A[i-1] = 0$ OR $A[i-1] \cdot 10 + A[i] > 26$.
$d[i] = d[i-2]$ if $A[i] = 0$.
$d[i] = d[i-1] + d[i-2]$ Otherwise.

Compute $d[i]$ in increasing order of i. The total number of ways to decode A is $d[n]$.

Note that the encoded message is encoded from some letters, so there must be some ways to decode it.

Student ID: ________

5. **Dynamic Programming [10 pts]**

Given a $n \times m$ binary matrix A filled with 0's and 1's, find the area of the largest square containing all 1's in $O(nm)$ time.

For example, given the following matrix:

$$\begin{array}{ccccc} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{array}$$

The algorithm returns 4.

Solution: For $1 \le i \le n$ and $1 \le j \le m$, define $s[i,j]$ be the area of the largest square containing all 1's of the submatrix $A[1..i, 1..j]$ where the square contains $A[i,j]$.

Base case: $s[1][j] = A[1][j]$ for $1 \le j \le m$, $s[i][1] = A[i][1]$.
Recursive case: $s[i,j] = \min\{s[i-1,j], s[i,j-1], s[i-1,j-1]\} + 1$ if $A[i,j] = 1$. Otherwise, $s[i,j] = 0$

$s[i,j]$ is computed row by row in increasing order of i and increasing order of j. The area of the largest square containing all 1's of the submatrix $A[1..i, 1..j]$ is $\max_{i,j}\{s[i,j]\}$.

Student ID: ________

6. **Dynamic programming [10 pts]**
A sequence of numbers $a_1, a_2, a_3, \ldots, a_n$ is *oscillating* if $a_i < a_{i+1}$ for every odd index i and $a_i > a_{i+1}$ for every even index i. For example, the sequence 2, 7, 1, 8, 2, 8, 1, 8, 3 is oscillating. Describe and analyze an efficient algorithm to find a longest oscillating subsequence in a sequence of n integers. Your algorithm only needs to output the length of the oscillating subsequence. For example if the input sequence is 2, 4, 5, 1, 4, 2, 1, your algorithm should output 5, corresponding to the subsequence 2, 4, 1, 4, 1, or 2, 4, 1, 4, 2, or any other such subsequence. For full credits, your algorithm should run in $O(n^2)$ time.

Solution: Let $o[i]$ be the length of the longest oscillating subsequence that ends at a_i and has an odd length; let $e[i]$ be the length of the longest oscillating subsequence that ends at a_i and has an even length. We have the recurrence: $o[1] = 1, o[i] = \max\{0, \max_{j<i, a_j>a_i} e[j]\} + 1, e[i] = \max\{-\infty, \max_{j<i, a_j<a_i} o[j]\}+1$. Then we compute $o[1], e[1], o[2], e[2], \ldots$. The final answer is $\max_i\{o[i], e[i]\}$.

Student ID: ________

7. **Minimum spanning tree [7 pts]**
Assume an undirected graph $G(V, E)$, where the number of nodes is a power of 2. Consider the following divide and conquer algorithm for computing a MST:

   1. Divide the nodes in V in two disjoint sets V_1 and V_2 of equal size. Let E_1 be the set of edges that are incident only on vertices in V_1, and E_2 be the set of edges that are incident only on vertices in V_2.
   2. Recursively compute a MST T_1 in $G_1(V_1, E_1)$ and a MST T_2 in $G_2(V_2, E_2)$.
   3. Connect T_1 and T_2 with the lightest edge crossing the cut V_1, V_2, i.e., the edge with the minimum weight between a node in V_1 and a node in V_2.

   Either show that the resulting tree is a MST of $G(V, E)$, or describe a counter example where the algorithm fails.

Solution: The algorithm fails. Consider nodes $\{1, 2, 3, 4\}$ and edges $(1, 2), (2, 3), (3, 4), (1, 4)$ with weights 10,1,2,20, respectively. If we set $V_1 = \{1, 2\}, V_2 = \{3, 4\}$, then $T_1 = \{(1, 2)\}$ (weight 10), $T_2 = \{(3, 4)\}$ (weight 10). The output of the algorithm will be $\{(1, 2), (3, 4), (2, 3)\}$ with total weight 21. The MST is $\{(1, 2), (2, 3), (3, 4)\}$ with weight 13.

Student ID: ________

8. **Minimum spanning tree and Shortest path [8 pts]**
Let T be a MST of an undirected graph G. For each of the following statements, state if it is true or false. If true, explain why. If false, give a counter-example.

(a) For any two nodes s, t, the $s-t$ path in T is a shortest $s-t$ path in G.

(b) For any two nodes s, t, the $s-t$ path in T is the $s-t$ path in G that minimizes the weight of the heaviest edge in the path.

Solution: (a) False. Consider $w(s,u) = 2, w(u,t) = 3, w(s,t) = 4$. T contains $(s,u), (u,t)$ and the $s-t$ distance is 5. In the original graph, the shortest $s-t$ distance is 4.

(b) True. Assume that (u,v) is the heaviest edge in the $s-t$ path. removing (u,v) from T creates two disconnected components, one containing s, u, and the other containing v, t. (u,v) must be the lightest edge crossing the cut. Thus, any other edge connecting s and t has weight at least $w(u,v)$.

Student ID: __________

9. **All Pairs Shortest Paths [10 pts]**
Assume an undirected connected graph $G(V, E)$, where all edges have the same weight w where $w > 0$. Give an $O(VE)$ algorithm for computing the shortest distance between all pairs of nodes.

Solution: This is the same as finding the path with minimum number of edges between all pairs of nodes. This can be done by applying BFS on G for $|V|$ times, where each vertex has been the source for one time. For any two nodes u, v, the shortest distance is the w multiply the number of edges in the path we have found for u, v. The running time is $O(V(V + E)) = O(VE)$, since the graph is connected.

Student ID: ________

10. **Maximum Flows [12 pts]**
Consider a directed graph $G(V, E)$, where all edge capacities equal 1. Assume that the maximum flow from the starting node s, to the terminal node t is f. Given an integer k $(0 < k < f)$, describe an algorithm to select k edges to delete from G, in order to reduce the maximum $s - t$ flow as much as possible. In other words, we want to minimize the maximum flow in the new graph, after the removal of the edges. Discuss the value of the maximum flow in the new graph and the complexity of your algorithm.

For full credits, your algorithm should run in $O(VE^2)$.

Solution: Find the maximum flow in G by Edmonds-Karp algorithm. When the algorithm terminates, there is an $S - T$ cut, where the set S contains the nodes reachable from s. The total capacity of edges from S to T equals the value of the maximum flow f. Since each edge has capacity 1, there are $f(> k)$ such edges. Select k of these edges to remove. The max flow in the new graph has value $f - k$.

Edmonds-Karp algorithm runs in $O(VE^2)$. The k edges cross the cut can be found as following. Performing BFS from s in the residual graph. All nodes reachable belong to set S. Find the edges from S to T by scanning the adjacency lists of the nodes in S. This can be done in $O(E)$ time. So, the total cost is dominated by Edmonds-Karp algorithm, which is $O(VE^2)$.

Student ID: ________

11. **Stable Marriage [6 pts]**
Consider 3 men M_1, M_2, M_3 and three women W_1, W_2, W_3 with the following preferences:

For men

M_1:	W_1,	W_2,	W_3
M_2:	W_1,	W_2,	W_3
M_3:	W_2,	W_1,	W_3

For women

W_1:	M_1,	M_2,	M_3
W_2:	M_2,	M_1,	M_3
W_3:	M_2,	M_3,	M_3

(a) Show that the assignment $(M_1, W_1), (M_2, W_3), (M_3, W_2)$ is unstable.

(b) Find a stable matching and explain whether it is optimal for the men, the women or both.

Solution: (a) The pair (M_2, W_2) is unstable because they would prefer to be together than their currents partners (M_2, W_3) and (M_3, W_2).

(b) The stable matching is $(M_1, W_1), (M_2, W_2), (M_3, W_3)$.
(M_1, W_1) can only be matched to each other because each is first in the preference list of the other.
(M_2, W_2) must be matched together, otherwise they are an unstable pair.
Thus, (M_3, W_3) must be the third pair. This is the only matching possible, and is optimal for both men and women.

Student ID: ________

Scrap Paper

Student ID: ________

Scrap Paper