CSci 242: Algorithms and Data Structures

Instructor: Dr. M. E. Kim

Date: December 13th, 2018

Final Exam: 150 points + 30 (Optional)

Name: ________________________________

1. Your answer should be precise and fully described; any sloppy answer will not get a full point.
2. Your hand writing should be **clear, dark, large enough (≥ 9 pt.)** and **readable**.

Mark the followings;

Difficulty:

Very Easy: _____ Easy: _____ Moderate: _____ Difficult: _____ Very Difficulty: _____

Time:

Short: ________ Enough: ________ Too Much: _________

Comment:

Q1. [30] **True or False**

Justify your answer clearly. Give a counter example if necessary.

(1) [2+3] In the Kruskal's algorithm to find the Minimum Spanning Tree, the Priority Queue maintains the vertices of graph in order to get the vertex with the minimum weight edge.

(2) [2+3] A large *minimum degree B* of B-tree, where B is the number of items that can fit in one block, increases the number of disk access required to find a key.

(3) [2+3] Quick sort is much faster than quadratic sorting algorithm such as insertion-sort and selection-sort.

(4) [2+3] Dijkstra's algorithm is successful to find the shortest path from a given source in the graph which contains a negative weight edge.

(5) [2+3] The optimal data encoding/decoding can be *always* achieved by a variable length code word with no ambiguity.

(6) [2+3] A greedy algorithm makes a greedy choice before finding an optimal solution of sub- problem. Thus, it solves a problem in top-down approach.

Q2. [40] **Short Answer**

Explain or define your answer clearly.

(1) [5] Define a (a, b)-*tree* and explain its property.

(2) [5] Give the maximum height and the minimum height of (a, b)-tree in the asymptotic notation O and Ω, respectively.

(3) [5] For a B-tree T of the maximum degree b and the minimum degree $a=\lceil b/2 \rceil$, what is (A) the minimum number of items and (B) the maximum number of items that a root can store, respectively?

(4) [5] Define a strongly connected component of a graph G = (V, E).

(5) [10] For a given hash table of size 11, insert the keys 16 where a collision is handled by ***quadratic probing***.
The main hash function is ***h(k)=k*** **mod 11** and the **secondary quadratic function** $\boldsymbol{f(j) = 3j^2 + j}$.
Show the proper computational steps.

0	1	2	3	4	5	6	7	8	9	10
					5			30	9	21

(6) [5] Describe 4 types of edge in Depth First Search in Directed Graph.

(7) [5] For a tree stored in the given array, build a *maximum* heap. Show the final heap in the array.
Do not use 'Bottom-Heap' construction, but by performing 'UpHeap' operation from j = 1 to n.

1	9	7	12	5	6	3	10	4	2

Q3. [20] Fractional Knapsack Problem

An edited book has 8 articles. The table shows the lengths of the articles and their importance, where the scale of importance is 1(low) to 10(high). The book must be at most 200 pages long. The problem is ***to edit the book*** so that **the overall importance is maximized.**

(1) [15] Edit the book by choosing articles whose pages and importance are given in the table, giving

Article	Importance of article	Pages
A	3	15
B	5	20
C	6	20
D	4	10
E	3	30
F	3	90
G	4	60
H	6	40

(A) [5] the list of the chosen articles with their chosen number of pages in the order,

(B) [5] the importance of each chosen article, and

(C) [5] the total maximum importance of the edited book of 200 pages.

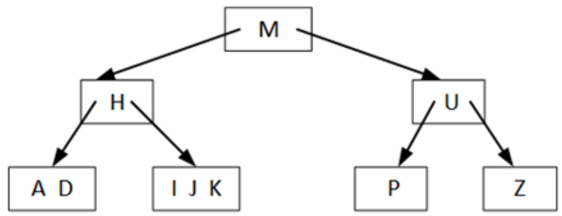
(2) [5] Consider a greedy rule for the above Fractional Knapsack Problem that selects the articles in non-decreasing order of pages. If the capacity of the knapsack is not exceeded, we take all of the pages. Otherwise, we take whatever portion of the article fills the knapsack and stop.
Give an example to show that this greedy algorithm does not necessarily maximize the importance.

Q4. [20] **B-Tree**

(1) [14] For a given B-tree whose order is 4, show the changes of B-tree

(A) after deleting a key ***P***, then,

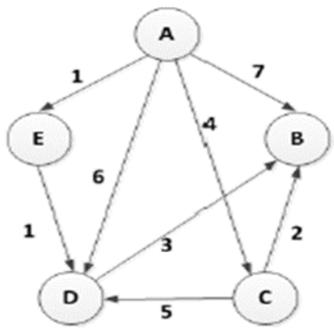
(B) inserting a key ***L*** to the B-tree in (A). Show the deletion & insertion step by step.



(2) [6] Describe 3 *page replacement strategies* in *blocking* if the primary memory is full when data is transferred from the external memory

Q5. [20] **Single Source Shortest Path in the DAG**

(1) [10] For a given DAG, (A) sort the vertices in the topological order and (B) redraw the graph by arranging the vertices in the sorted order

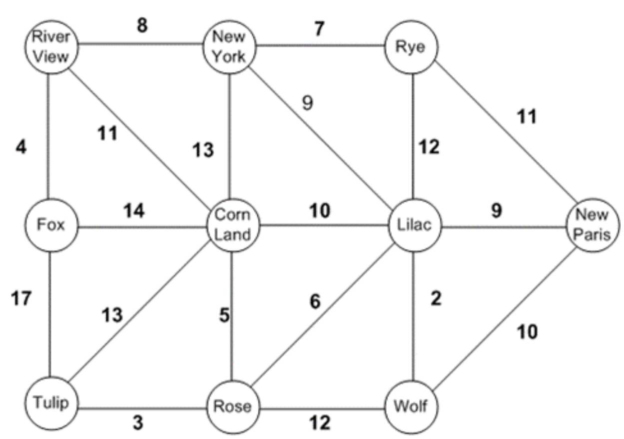


(2) [10] Find the shortest path from a vertex *A* to each vertex in (1). You have to show the proper steps of edge relaxations, updating a key, D[*v*] of each vertex *v*.

Q6. [20] **Minimum Spanning Tree (MST)**

(1) [10] Write Kruskal's algorithm in the pseudo code to find the *Minimum Spanning Tree*.

(2) [10] By applying Kruskal's algorithm to the given graph, (A) find its MST and (B) give the *total minimum weight*. You should show proper *changes of clusters,* the MST in growing, the smallest weight edge that cross the clouds.



Q7. [20, Optional] **Huffman Codes**

(1) [10] The text contains only the characters in the table with the given frequency. (A) Construct a Huffman Tree to generate and (B) give the optimal Huffman code for each character.

Character	Frequency	Huffman Code
Space	25	
A	16	
E	20	
I	12	
O	4	
G	10	
N	24	
R	18	
S	30	
T	40	
'	1	

(2) [5] What is the total number of bits required to encode the text using your Huffman codes?

(3) [5] Decode the following codes into a text.

11001011101101011010111011001101001011111110100100010100111011110

Q8. [10, Optional] **Recurrence**

Solve the following recurrence by one of the following methods:
Master's Theorem, Iterative Substitution or Recursion Tree method.

$$T(n) = \begin{cases} O(1) & n < 4 \\ 3 \cdot T\left(\frac{n}{4}\right) + O(n) & n \geq 4 \end{cases}$$

You should clearly state the followings, depending of the method of your choice

- Master's Theorem :
  the case to which it belongs, the rationale of the solution and the solution in the asymptotic bound,
- Iterative Substitution:
  The proper number of iterative steps and the final step in which the number of input is 1, the computation of some parameters.
- Recursion Tree :
  the height of tree, the number of leaves, level, the number of nodes per level, the size of input per level, a per-level time, the total time **and** its asymptotic bound (i.e. the asymptotic bound of solution),