School of Computer Science
University of Guelph

CIS*3490 The Analysis and Design of Algorithms

Winter 2021
Instructor: Fangju Wang

Assignment 5 (100%)

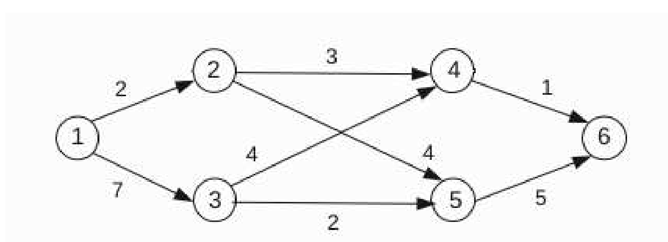
Question 1 (20%)

1.1 Construct an AVL tree for list 35, 42, 79, 18, 23, 38. Insert the keys into the tree in the order of the list. For each key, show the tree with balance factors after the key is inserted, and the resultant tree if a rotation is conducted.

1.2 Apply the method of heapsort to sort list 23, 35, 42, 79, 18, 38 in ascending order. Do the sorting in the array representation. Start with the original order of the numbers. Show steps in both the stage of heap construction and stage of sorting. Please don't draw trees in your submission for this question.

Question 2 (20%)

1.1 Apply the shortest augmenting path algorithm to find the maximum flow in the following network. Break a tie using the numerical order. For each augmenting path you find, draw the network, label the vertices, list vertices of the path, and calculate the flow value of the path, using the format of Figure 10.7 (page 368) in the text. When you have found all the paths, calculate the maximum flow.



2.2 Find a stable marriage matching for the instance defined by the following ranking matrix. Break a tie using the alphabetic order. Draw a matrix after each iteration, and describe what happens in the iteration, using the format of Figure 10.12 (page 382). When the process is over, write the matching.

	A	B	C	D
α	$1,3$	$2,3$	$3,2$	$4,3$
β	$1,4$	$4,1$	$3,4$	$2,2$
γ	$2,2$	$1,4$	$3,3$	$4,1$
δ	$4,1$	$2,2$	$3,1$	$1,4$

Question 3 (20%)

3.1 For each question in the following, find which item is NOT true. (Choose the most appropriate answer. No penalty for incorrect answers.)

1. The knapsack problem
   (a) is an NP-complete problem.
   (b) is an NP problem.
   (c) is a P problem because it has a dynamic programming algorithm of $\Theta(nw)$ where n is the number of items and w is the capacity of knapsack.
   (d) can be polynomially reducible to an NP-complete problem.

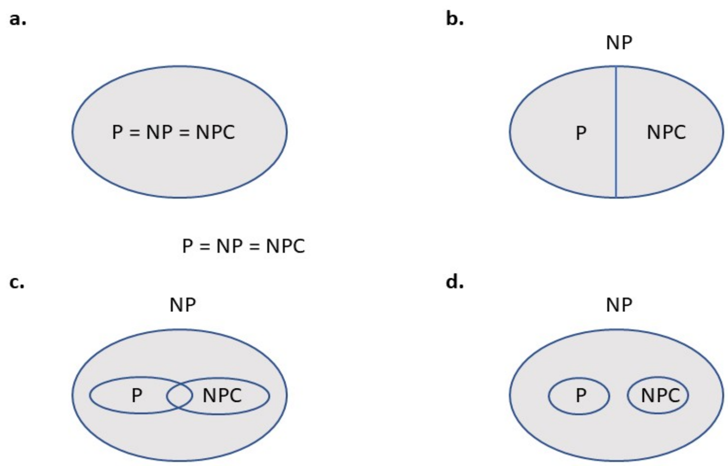
2. $P = NP$
   (a) holds if a P algorithm has been found for an NP-complete problem.
   (b) implies that we have a P algorithm for every NP problem.
   (c) has been disproved by a scientist at HP.
   (d) can be proved by finding a P algorithm for the CNF-satisfiability problem.

3. An *NP* complete problem can be solved by
   (a) an exponential or factorial algorithm.
   (b) a deterministic polynomial algorithm.
   (c) a nondeterministic polynomial algorithm.
   (d) an algorithm that consists of a guess stage and a verification stage.

4. An *NP* complete problem
   (a) is an *NP* problem.
   (b) is in *NP* and at least as difficult as any other problem in *NP*.
   (c) is an *NP* problem that has been completed.
   (d) is an *NP* problem and any other problem in *NP* can be reduced to it in polynomial time.

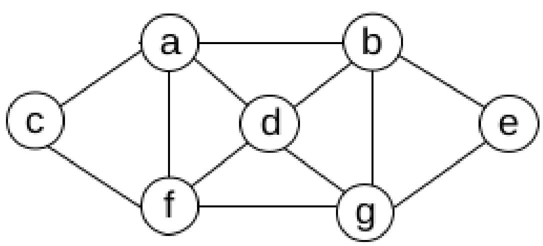
5. The class *NP* includes the decision versions of:
   (a) assignment problem.
   (b) traveling salesman problem
   (c) Fibonacci number calculation
   (d) bin packing.

3.2 Which of the following diagrams do not conflict the current state of our knowledge about the complexity classes P, NP, and NPC (NP-complete problem)? That is, which of the diagrams are possible, and which are impossible? Explain your answer for each diagram using one or two sentences.

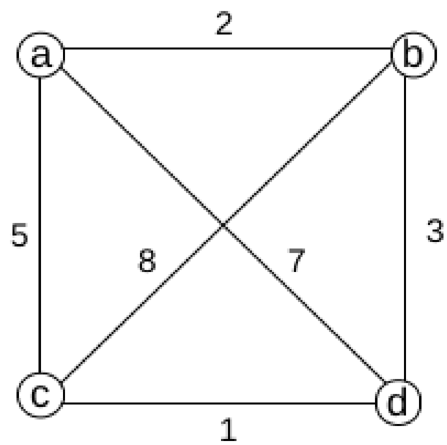


Question 4 (20%)

4.1 Apply backtracking to the problem of finding a Hamiltonian circuit in the following graph. Draw your state-space tree in the format of Figure 12.3 (page 427).



4.2 Apply the branch-and-bound algorithm to solve the traveling salesman problem for the following graph. Draw your state-space tree in the format of Figure 12.9 (page 439).



Question 5 (20%)

5.1 The first-fit decreasing (FFD) approximation algorithm for the bin packing problem starts by sorting the items in non-increasing order of their sizes and then acts as the first-fit algorithm. Apply FFD to the instance when bin size is 1.0.

$$s_1 = 0.4,\ s_2 = 0.7,\ s_3 = 0.2,\ s_4 = 0.1,\ s_5 = 0.5.$$

Is the solution obtained optimal? Why?

5.2 The following is an instance of the knapsack problem, where knapsack capacity is 11. Use the Sahni's approximation scheme to solve this knapsack problem for $k = 2$. Show your process using the format of Figure 12.16 (b) (page 457).

Item	*weight*	*value*
1	7	\$42
2	3	\$12
3	4	\$40
4	5	\$25

Due time: 23:59, Monday, April 12, 2020. Please submit an e-copy to Moodle.