

CS 251 Homework 4: Data Structures and Algorithms (100 points)

Out: April 5, 2019 (8:00 pm)

Due: April 12, 2019 (8:00 pm)

Name: Liyao (Mars) Gao

Instructions: Each **multiple choice** question has only one correct answer. Additionally, you must provide an **explanation** for your answer (2-3 short sentences is sufficient). Answers without proper explanations, even though it is correct, will be graded with 0 points.

For **true/false** questions, you must answer true or false AND explain your reasoning.

For all other questions, follow the specific instructions carefully.

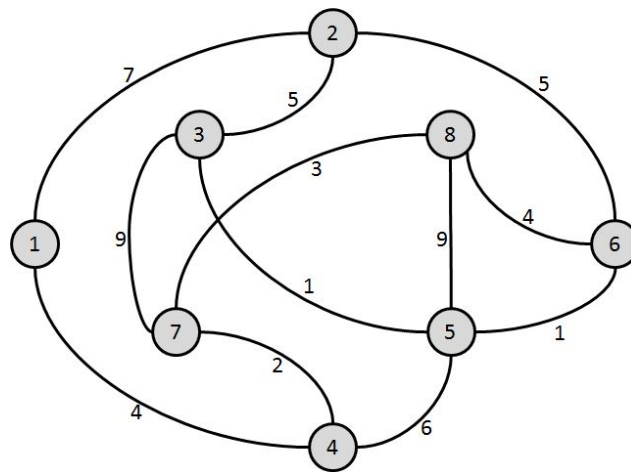
For all questions, put your responses in the boxes provided.

1. (10pts) Given a dense directed graph $G(V,E)$ with possibly negative weights (but no negative cycle), for each of the following scenarios very briefly explain how to use one of the fastest algorithms (in the lectures) and write down its time complexity.

- a) Find cycles
- b) All-pairs shortest paths
- c) Single-source shortest paths
- d) Minimum spanning tree

- a) Cycle DFS.
- b) Floyd Warshall. $O(n^2)$
- c) Dijkstra.
- d) Prim's

2. (10pts) Use Prim's algorithm to find the minimum spanning tree of the following graph. Start from vertex 8 and list the edges in the order in which they are chosen.




[(8, 7), (7, 4), (8, 6), (6, 5), (4, 1), (6, 2), (2, 3)]

Visited List: [8, 7, 4, 6, 5, 1, 2, 3]

3. (10pts) Using Huffman tree encoding, find an efficient code for a file of 115 characters and the following frequency table. Draw your tree and label the edges as follows:

1. For each node put its smaller sub-tree on the left.
2. Left-edges are numbered 0 and right-edges are numbered 1.

a	b	c	d	e	f	g
24	15	16	11	5	37	7



4. (10pts) Let (u,v) be a minimum-weight edge in a connected graph G . (Exclude the case in which all edges have equal weights). Write an argument that (u,v) belongs to some minimum spanning tree of G . (Hint: Prove by contradiction. Take an arbitrary MST [i.e. T] and argue that you can find another tree whose weight is less than T .)

Suppose a minimum spanning tree of G is T .

i) If u is v 's parent, grandparent, branch. If u, v are not adjacent in T .

There exist a subset of vertices, $a = \{a_1, a_2, \dots, a_n\}$, $|a| < \text{number of vertices in } T$, such that $[u \rightarrow a_1 \rightarrow a_2 \rightarrow \dots \rightarrow v]$.

The sum of these edges are $w(u, a_1) + w(a_1, a_2) + \dots + w(a_n \rightarrow v)$.

As an alternative method to connect, we can make $[u \rightarrow v \rightarrow a_n \rightarrow \dots \rightarrow a_1]$.

The sum of these edges are $w(u, v) + w(v, a_n) + \dots + w(a_2 \rightarrow a_1)$

$= w(u, v) + w(a_1, a_2) + \dots + w(a_n \rightarrow v)$

Since $w(u, v)$ is the smallest edge,

$w(u, v) + w(v, a_n) + \dots + w(a_2 \rightarrow a_1) < w(u, a_1) + w(a_1, a_2) + \dots + w(a_n \rightarrow v)$.

T is not MST. Contradiction.

ii) If v is u 's parent, grandparent, branch. If u, v are not adjacent in T .

For the same reason, contradiction.

iii) If u, v , have same parent, grandparent, ... Suppose their parent is p_u, p_v .

Consider only $(p_u, u), (p_v, v)$.

$\text{Sum}_1 = w(p_u, u) + w(p_v, v) + K$, where K is the weights of other edges.

If we change the edges to $(p_u, u), (u, v)$, where other edges remain the same,

$\text{Sum}_2 = w(p_u, u) + w(u, v) + K$.

Since $w(u, v)$ is the smallest edge, $\text{Sum}_2 < \text{Sum}_1$.

T is not MST. Contradiction.

5. (10pts) In a text of length 1000 with the following pattern

1234123412341234...

How many successful comparisons and how many unsuccessful comparisons are made by the brute-force string-matching algorithm to search for all occurrences of 341? We are looking for two answers here. (A successful comparison is when one character from the pattern matches one character from the text, otherwise it is unsuccessful.)

When length = 4,

1234 has no 341. 0 occurrence.

1234 1234

341 341

fail fail

Fail: 2, Success: 0.

When length = 8,

12341234 has 1 occurrence.

12341234 12341234 12341234 12341234 12341234 12341234

341 341 341 341 341 341

fail fail success fail fail fail

Fail: 5, Success: 1.

$|Fail| + |Success| = \text{length} - 2$

$|Success| = (\text{length} - 4)/4$

So when length = 1000,

$|Success| = 249$

$|Fail| = 749$

6. (10pts) Write the pseudocode of an algorithm to compute Last-Occurrence Function in time $O(m + |\Sigma|)$, where m is the size of pattern P and Σ is the set of symbols.

7. (10pts) Use Boyer-Moore's algorithm to find the substring "puppet" in the text "puppy puppet looks happy".

(a). First compute the last occurrence function. For the alphabet assume it is the letters that compose the text. Ignore blank spaces.

(b). Following the Boyer-Moore algorithm, what is the number of comparisons required for finding the pattern in the given text? Include a trace of the algorithm execution, similar to those shown in class, positioning puppet beneath the beginning of the text and then at all its subsequent positions until found in the text.

8. (10pts) Suppose that all characters in pattern P are different. Show the pseudocode of an algorithm that accelerates the brute-force algorithm to run in time $O(n)$ on a text of n characters.

9. (10pts) Consider the following algorithm for finding a minimum spanning tree in a connected weighted graph (w is considered as the weight function). Determine if the output of this algorithm is always a minimum spanning tree of the graph or not and explain how you know. (Hint: Recall the cycle property discussed in class.)

sort the edges into non increasing order ($w(e_1) \geq w(e_2) \cdots \geq w(e_m)$)

Initialize T to be equal to the set of all edges E

for $i = 1$ to m **do**

if $T - \{e_i\}$ is a connected graph **then**

$T = T - \{e_i\}$

end if

end for

return T

10. (10pts) Finding patterns in DNA sequences is a common task in bioinformatics. A DNA sequence is composed of characters A, C, G and T representing adenine, cytosine, guanine and thymine respectively. The KMP algorithm is used when the text and the pattern are not too long. Before running KMP we must calculate the failure function of the pattern. We need to find the pattern "GACAGATGA" in a DNA sequence.

(a) Calculate the failure function for the given pattern.

(b) Following the KMP algorithm, what is the number of comparisons required for finding the pattern "GACAGATGA" in "GGTACCCGACAGATGACAGA"? Include a trace of the algorithm execution, similar to those shown in class, positioning the pattern beneath the beginning of the text and then at all its subsequent positions until found in the text.