Part I: Analysis and Design of Algorithms

*Answer as much as you can. Maximum grade = **50 pts.***

Question 1 [**10 Marks**] Show how to multiply two linear polynomials $ax + b$ and $cx + d$ using **only three** multiplications. (*Hint*: One of the multiplications is $(a + b) \cdot (c + d)$.) Describe an efficient algorithm for multiplying two polynomials of degree n. What is the running time of your algorithm?

Question 2 [**10 Marks**] Describe a dynamic-programming solution to the 0–1 knapsack problem that runs in O(nW) time, where n is number of items and W is the maximum weight of items that the thief can put in his knapsack.

Question 3 [**15 Marks**] Consider the following algorithm:

```
MAYBE-MST(G, w)
  T ← Ø
  for each edge e of G, taken in arbitrary order
  do T ← T ∪ {e}
     if T has a cycle c
       then let é be the maximum-weight edge on c
            T ← T - {é}
  return T
```

a. **[5 marks]** Does the algorithm compute a minimum spanning tree? Why?
b. **[5 marks]** What is the running time of the **most efficient** implementation of this algorithm? Why?
c. [**5 marks]** Describe an algorithm that resembles the above algorithm for efficiently computing the minimum spanning tree. What is its running time?

Question 4 [**15 Marks]** The input to the **subset-sum problem** is a pair *(S, t)*, where *S* is a set *{x1, x2, ..., xn}* of positive integers and *t* is a positive integer. This decision problem asks whether there exists a subset of *S* that adds up exactly to the target value *t*.
a. **[3 marks]** Is the subset-sum problem in P or in NP? Why?
b. **[6 marks]** Describe an exact algorithm to solve the subset problem. What is the running time of your algorithm?
c. **[6 marks]** Describe an efficient algorithm to solve the subset problem. What is the running time of your algorithm? Will your algorithm always give the correct answer? Why?

Question 5 [**15 Marks]** Shortest-path computation algorithms include Dijkstra's, Bellman-Ford, Floyd-Warshall, and Johnson's.
a. **[4 marks]** State the running time of each of the four algorithms.
b. **[2 marks]** Give a simple example of a directed graph with negative-weight edges for which Dijkstra's algorithm produces incorrect answers.
b. [**2 marks**] You are given an undirected graph and asked to compute **all-pairs** shortest paths on it. Describe a methodology to **select** the best algorithm to solve this problem. (*Hint*: select an algorithm based on input graph properties, such as edge count, vertex count, and values of edge weights.)
c. [**7 marks**] Describe **one** algorithm for computing all-pairs shortest paths. State its running time and prove its correctness.

PART II: THEORY OF COMPUTATION
ANSWER ALL QUESTIONS

1- Describe the language for each of the following regular expressions:
- a- a + b*
- b- ab* + bc*
- c- a*bc* + ac

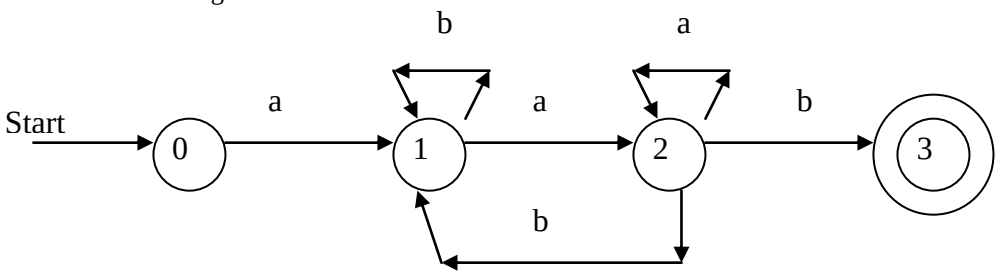
(5 points)

2- Find a regular expression for each of the following languages over the alphabet [a,b]:
- a- strings with even length
- b- strings whose length is a multiple of 3
- c- strings containing the substring aba
- d- strings with an odd number of a's.

(5 points)

3- Transform the following regular expression to NFA then to DFA: a(a+b)*.
(5 points)

4- Given the following NFA:

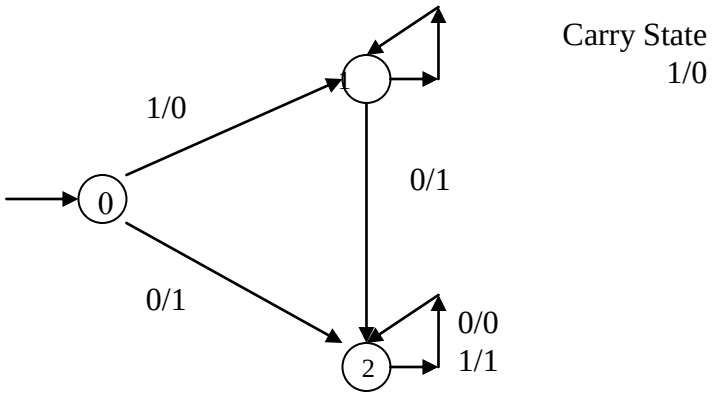


Find two regular expressions for the language accepted by the NFA as follows:
Delete state 1 before deleting state 2.
Delete state 2 before deleting state 1.
(5 points)

5- Find a context-free grammar for each of the following languages over the alphabet [a, b]
- a- $\{ a^n b^{2n} \mid n >= 0 \}$
- b- $\{ a^n b^{n+2} \mid n >= 0 \}$
- c- The palindromes of even length
- d- The palindromes of odd length

(5 points)

6- Given the Mealy machine for successor of a binary number:



a. compute the successor of 13
b. compute the successor of 7. Explain the result.
(5 points)

7- Transform the following PDA with empty stack acceptance and starting symbol X, into a context-free grammar:
<0, a, X, push(Y), 0>
<0, a, Y, push(Y), 0>
<0, b, Y, nop, 1>
<1, b, X, pop, 1>
<1, b, Y, pop, 1>
(5 points)

8- Given the following Context free grammar CFG:
S -> aSbc | lambda
Determine the relevant PDA by empty stack acceptance. Parse string 'aabcbc'
(5 points)

9- Construct a Turing Machine that adds 3 to a binary unary number. Note that in unary binary numbers 4=1111 and 3=111.
(5 points)

10- Write the Turing Machine instructions that adds a 1 to a binary natural number. The algorithm is as follows:
Move to the right end of the string
Repeat
If current cell contains 1, write 0 and move left
Until current cell contains 0 or lambda
Write a 1
Move to left end of string and halt
(5 points)