University of Toronto
Faculty of Applied Science and Engineering

CSC 190S

COMPUTER ALGORITHMS, DATA STRUCTURES AND LANGUAGES

Spring 1997

FINAL EXAM

Examiner: T.S. Abdelrahman

This test is open book(s) and open notes. Use of computing devices is NOT permitted.

Do not remove any sheets from this test book. Answer all questions in the space provided. No additional sheets are permitted.

Work independently. The value of each part of each question is indicated. The total value of all questions is 100.

Write your name and student number in the space below. Do the same on the top of each sheet of this test book.

Name (print): _____
(underline last name)

Student Number: _____

Q1._____ Q6._____

Q2._____ Q7._____

Q3._____ Q8._____

Q4._____ Q9._____

Q5._____ Q10._____

Total: [ ]

Question 1 (10 marks). *General.*

Answer the following questions by circling either True or False, or by providing a *brief* answer where appropriate.

1. Dijkstra's algorithm for finding the shortest path between two vertices in a graph works if there are negative weight edges, but not if there are negative weight cycles.

   True or False?

2. Although quadratic probing eliminates both primary and secondary clustering, it does not probe all the hash table locations, hence, underutilizing the table.

   True or False?

3. Heuristic search can solve any NP problem in polynomial time.

   True or False?

4. What is the minimum and maximum number of nodes in a heap of height h.

5. The set of all valid sentences in a language is the set T^*, where T is the set of terminals.

   True or False?

6. Which is a better search strategy for the 8-puzzle problem, breadth-first or depth-first?

A connected graph with n vertices and $n-1$ edges can have at most one simple cycle in it.

True or False?

8. In one sentence, explain how the heapsort program you implemented in assignment 4 can be modified to sort the array in decreasing order instead of increasing order.

9. Can hashing functions be made better by using a *truly* random number generator to map a key into a table location? Explain briefly.

10. Any algorithm for finding *all* connected components of a graph has exponential time complexity.

True or False?

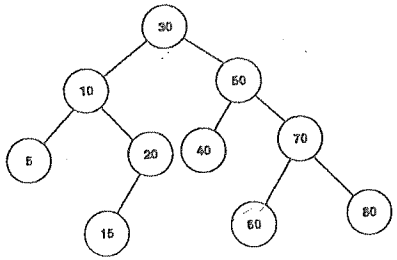
ion 2 (5 marks). *Complexity Analysis.*

mine the time complexity of the following program segment as a function of n, the input size.

```
for i : 1 .. n
  for j : 1 .. i*n
    O(1)
  end for
end for
```

Question 3 (8 marks). *Balanced Trees.*

The tree shown below is an AVL tree.

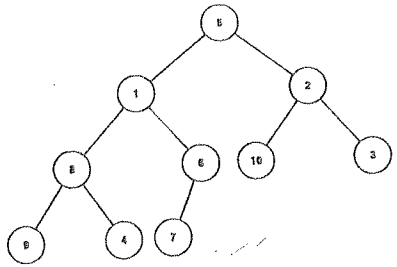


a. (3 marks). Insert a node with the key 90 on the tree and redraw the tree below without re-balancing.

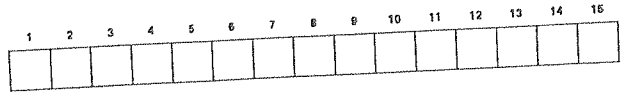
b. (5 marks). If the tree in part a. is imbalanced, then re-balance it and re-draw the resulting balanced tree below. Indicate the type of rotation (if any) that you used to re-balance the tree (e.g., double right rotation, etc).

Question 4 (12 marks). *Heaps.*

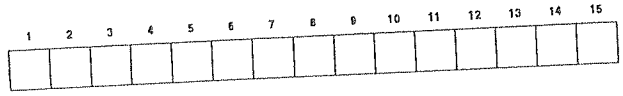
The following is a binary tree.



a. (3 mark). Draw the binary tree using its array representation.



b. (3 marks). Build a heap from the elements of the array. Show **both** the array and tree representation of the heap after the build is done.



rks). Start with the heap in part b. and insert a new element with the key "15" and restore
if necessary. Show only the tree representation of the heap after the heap is restored.

(3 marks). Start with the heap in part b. and delete the element with the largest key then restore
heap as needed. Show only the tree representation of the heap after the heap is restored.

Question 5 (15 marks). *Hash Tables.*

Assume a hash table with size $T = 13$, and the following hash function:

index = (key **mod** T) + 1.

a. (3 marks). Show the contents of the hash table after the following operations have been performed. Indicate next to each operation the number of *probes* performed. Assume that linear probing is used to resolve collisions and that the hash table is initially empty.

1. insert 7.
2. insert 47.
3. insert 20.
4. insert 35.
5. insert 23.
6. insert 59.

index	key
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	

b. (3 marks). Redo part a., but using quadratic probing to resolve collisions. Again, indicate next to each operation the number of *probes* performed and assume that the hash table is initially empty.

1. insert 7.
2. insert 47.
3. insert 20.
4. insert 35.
5. insert 23.
6. insert 59.

index	key
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	

c. (3 marks). Redo part a., but using double hashing to resolve collisions. Use the following function as the second hashing function:

index = (key mod (T-2)) + 1.

If collisions persist, resolve them using linear probing. Again, indicate next to each operation the number of *probes* performed and assume that the hash table is initially empty.

index	key
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	

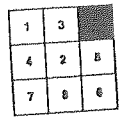
1. insert 7.
2. insert 47.
3. insert 20.
4. insert 35.
5. insert 23.
6. insert 59.

d. (3 marks). For part b. above, identify the first insert operation for which quadratic probing reduced the number of probes by eliminating *primary* clustering.

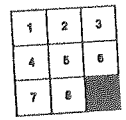
e. (3 marks). For part c. above, identify the first insert operation for which double hashing reduced the number of probes by eliminating *secondary* clustering.

Question 6 (10 marks). *Heuristic Search.*

Show the search tree generated by best-first search to solve the following 8-puzzle problem. Be sure to label each node in the tree with its merit value.



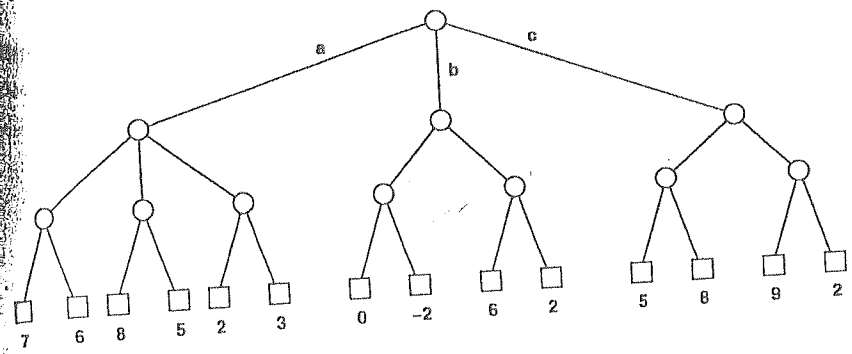
Start



Goal

stion 7 (8 marks). *Game Trees.*

t of the tree for a fictitious game is shown below. The value returned by the evaluation function plied to each leaf of the tree is indicated next to the leaf.

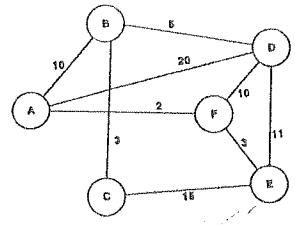


a. (4 marks). Using the minmax algorithm, indicate which move should be made by the machine (a, b or c).

b. (4 marks). Circle the parts of the tree that would be pruned had alpha-beta search been used instead. Assume the search is done depth-first left-to-right. Which move should be made by the machine in this case?

stion 8 (13 marks). *Graphs.*

wn below is an undirected weighted graph G.



a. (6 marks). Give the breadth-first and depth-first traversals of G. Start at vertex A. In the case of he depth-first traversal, use the stack-based algorithm described in class.

Breadth-first:

Depth-first:

b. (3 marks). Show the spanning tree generated by the breadth-first traversal by highlighting the edges of the spanning tree on the graph above.

c. (4 marks). If the spanning tree obtained in part b. above is not a minimal spanning tree for the graph, then obtain the minimum spanning tree and draw it below.

estion 9 (10 marks). *Divide and Conquer.*

Canadian currency has coins in 1-cent, 5-cent, 10-cent, 25-cent, loonie (one dollar) and toonie (two dollars) denominations. We can easily make change using the following *greedy* algorithm which also minimizes the number of coins used: *repreatedly use the largest coin possible*. For example, one can make $1.33 in change by using six coins: one loonie, one 25-cent, one 5-cent, and three 1-cent.

a (2 marks). The above algorithm works only for the above denominations. In *Timbukta* (not to be confused with *Timbuktu*), there is a 21-cent coin in addition to the ones above. Show that in Timbukta the greedy algorithm fails to make change using the smallest number of coins.

b. (8 marks). Write a *recursive* algorithm (in pseudocode or C++) to make change with the smallest number of coins possible for any currency system with n coin denominations. The inputs to your algorithm are the amount of change in cents and an array called `denominations[n]` that contains the values of available coin denominations. The output is an array `change[n]` that gives the number of coins for each denominations. Form example, to make $1.33 in Canadian currency, the inputs are 133 cents and `[200 100 25 10 5 1]`. The output is `[0 1 1 0 1 3]`

Hint: consider an approach in which you select one coin, and then recursively determine the number of coins needed for the remainder.

Caution: Your solution should be *really* short (10 lines or so). Excessively long solutions will be penalized.

on 10 (9 marks). *Grammars and Parsers.*

narks). Write a grammar for the language whose sentences are *palindromes* consisting of a's only. Hence, `bab`, `aabbaa`, `bbbb` and `a` are all valid sentences in this language. Recall that drome is a word that reads forward the same as it reads in reverse.

marks). Using your grammar, show a derivation for the sentence `aabbaa`.

3 marks). Let $G = (T, N, S, P)$ be the following grammar:

$$\begin{aligned} T &= \{0, 1\} \\ N &= \{S\} \\ P &: S \longrightarrow 0S|1S|0|1|\epsilon \end{aligned}$$

Write a recursive-descent paring function `parse_S` for the language $L(G)$ generated by the grammar . You may assume two functions `current_token` and `next_token`, as defined in class, are available.