

Name: \_\_\_\_\_

Email address: \_\_\_\_\_

**CSE 373 Sample Midterm #2**  
(closed book, closed notes, calculators o.k.)

**Instructions** Read the directions for each question carefully before answering. We will give partial credit based on the work you **write down**, so show your work! Use only the data structures and algorithms we have discussed in class or which were mentioned in the book so far.

**Note:** For questions where you are drawing pictures, please circle your final answer for any credit. There is one extra page at the end of the exam that you may use for extra space on any problem. If you detach this page it must still be turned in with your exam when you leave.

**Advice** You have 50 minutes, **do the easy questions first**, and work quickly!

1) **True/False.** Circle True or False below. You do *not* need to justify your answers.

a. Linear probing is equivalent to double hashing with a secondary hash function of  $h_2(k) = 1$ . **True False**

b. If  $T_1(N) = O(f(n))$  and  $T_2(N) = O(f(n))$ , then  $T_1(N) = O(T_2(N))$ . **True False**

c. Building a heap from an array of  $N$  items requires  $\Omega(N \log N)$  time. **True False**

d. Merging heaps is faster with binary heaps than with leftist or skew heaps, because we only need to concatenate the heap arrays and then run BuildHeap on the result. **True False**

2) **Short Answer Problems:** *Be sure to answer all parts of the question!!*

a) Which ADT do binomial queues implement? If we forget simplicity of implementation, are they better than binary heaps? Are they better than leftist heaps? Justify your answer.

b) For node  $i$  in a ThreeHeap, give formulas that calculate each of the following:

(1) Node  $i$ 's parent

(2) Node  $i$ 's three children

c) What is the main advantage that open addressing hashing techniques have over separate chaining?

d) Fill in the blank in the following text to make the statement true.

In the union/find data structure of  $N$  items, if we use union-by-size *without path compression*, then any combination of  $M$  union and/or find operations takes at most \_\_\_\_\_ time.

### 3) Disjoint Sets:

Consider the set of initially unrelated elements 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12.

- a.) Draw the final forest of up-trees that results from the following sequence of operations using on union-by-size. Break ties by keeping the first argument as the root.

Union(0,2), Union(3,4), Union(9,7), Union(9,3), Union (6,8), Union(6,0),  
Union(12,6), Union(1,11), Union(9,6).

- b.) Draw the new forest of up-trees that results from doing a Find(4) with path compression on your forest of up-trees from (a).

**4) Heaps**

- a) Draw the leftist heap that results from inserting: 77, 22, 9, 68, 16, 34, 13, 8 in that order into an initially empty leftist heap. You do not need to show the array representation of the heap. You are only required to show the final heap, although if you draw intermediate heaps, ***please circle your final result for ANY credit.***

4 (cont.) **Heaps:**

- b) Draw the leftist heap that results from doing 2 deletemins on the heap you created in part a). You are only required to show the final heap, although if you draw intermediate heaps **please circle your final result for ANY credit.**

- c) What is the null path length of the root node in the last heap you drew in part b) above?

- d) List 2 good reasons why you might choose to use a skew heap rather than a leftist heap.

5) **Binomial Queues** –

a) What is the minimum and maximum number of nodes in a Binomial Tree of height  $h$ ?

b) What is the minimum and maximum number of nodes in a Binomial Queue whose tallest tree is of height  $h$ ?

c) Briefly describe how `Findmin()` is implemented for Binomial Queues.

6) Draw the contents of the hash table in the boxes below given the following conditions:

The size of the hash table is 12.

Open addressing and double hashing is used to resolve collisions.

The hash function used is  $H(k) = k \bmod 12$

The second hash function is:  $H_2(k) = 7 - (k \bmod 7)$

What values will be in the hash table after the following sequence of insertions? Draw the values in the boxes below, and show your work for partial credit.

33, 10, 9, 13, 12, 45, 26, 17

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	





## 8) B-trees

a) Given the following parameters:

Disk access time = 1 milli-sec per byte

1 Page on disk = 1024 bytes

Key = 16 bytes

Pointer = 4 bytes

Data = 128 bytes per record (includes key)

What are the best values for:

M =

L =

b) Insert the values 1, 2, 5, 4, 6, 3, 7, 8, 9 in that order into a B tree with  $L=2$  and  $M=3$ .