COS 226	Algorithms and Data Structures	Spring 2003

Midterm

This test has 7 questions worth a total of 50 points. You have 80 minutes. The exam is closed book, except that you are allowed to use a one page cheatsheet. No calculators or other electronic devices are permitted. Give your answers and show your work in the space provided. **Write out and sign the Honor Code pledge before turning in the test.**

"I pledge my honor that I have not violated the Honor Code during this examination."

Problem	Score
1	
2	
3	
4	
Sub 1	

Problem	Score
5	
6	
7	
Sub 2	

Total	

Name:

Login ID:

Precept:

1	12:30	Kevin
2	1:30	Adriana
3	3:30	Kevin

1. **Sorting algorithms. (10 points)**

   The column on the left is the original input of strings to be sorted. The columns to the right are the contents at some intermediate step during one of the 10 sorting algorithms listed below. Match up each algorithm by writing its number under the corresponding column. Use each number exactly once.

now	ace	ace	sob	ace	and	and	wee	ace	and	ace
for	dim	ago	nob	ago	ago	for	was	bet	for	ago
tip	for	and	cab	and	cue	dug	wad	dim	dug	and
ilk	hut	bet	wad	bet	ace	ilk	sob	egg	ilk	bet
dim	ilk	cab	and	cab	bet	dim	sky	few	dim	cab
tag	jot	caw	ace	caw	caw	ago	tip	for	ago	caw
jot	nob	cue	wee	cue	egg	fee	tar	hut	fee	cue
sob	now	dim	cue	dim	cab	cue	jay	ilk	cue	dim
nob	sky	dug	fee	dug	dim	caw	owl	jay	caw	dug
sky	sob	egg	tag	egg	for	cab	rap	jot	cab	egg
hut	tag	fee	egg	for	hut	hut	hut	men	hut	fee
ace	tip	few	gig	few	dug	ace	tag	nob	ace	few
bet	bet	now	dug	fee	gig	bet	cue	now	bet	ilk
men	men	for	ilk	gig	jam	gig	tap	sky	gig	men
egg	egg	tip	owl	hut	jot	egg	jot	sob	egg	sky
few	few	ilk	dim	ilk	fee	few	few	tag	few	now
jay	jay	gig	jam	jot	jay	joy	jam	tip	jam	jay
owl	owl	tag	men	jay	men	jam	nob	ago	owl	owl
joy	joy	jot	ago	joy	joy	jot	joy	and	joy	joy
rap	rap	sob	tip	jam	few	jay	now	cab	rap	rap
gig	gig	nob	rap	men	nob	men	gig	caw	men	gig
wee	wee	sky	tap	now	owl	wee	for	cue	wee	wee
was	was	hut	for	nob	tar	was	dim	dug	was	was
cab	cab	jam	tar	owl	ilk	sky	cab	fee	sky	sob
wad	wad	jay	was	rap	now	wad	ace	gig	wad	wad
caw	caw	men	jot	sob	sky	nob	caw	jam	nob	tag
cue	cue	joy	hut	sky	tip	sob	bet	joy	sob	jot
fee	fee	owl	bet	tip	rap	now	fee	owl	jot	hut
tap	tap	rap	now	tag	tap	tap	men	rap	tap	tap
ago	ago	tap	few	tap	tag	tag	ago	tap	tag	for
tar	tar	tar	caw	tar	was	tar	egg	tar	tar	tar
dug	dug	wad	sky	wee	sob	rap	dug	wad	tip	nob
and	and	was	jay	was	wad	tip	and	was	now	tip
jam	jam	wee	joy	wad	wee	owl	ilk	wee	jay	jam
---	---	---	---	---	---	---	---	---	---	---
0										

(0) Original input

(1) 3-way radix quicksort

(2) Bubble sort

(3) Heap sort

(4) Insertion sort

(5) LSD radix sort

(6) Mergesort

(7) MSD radix sort

(8) Quicksort

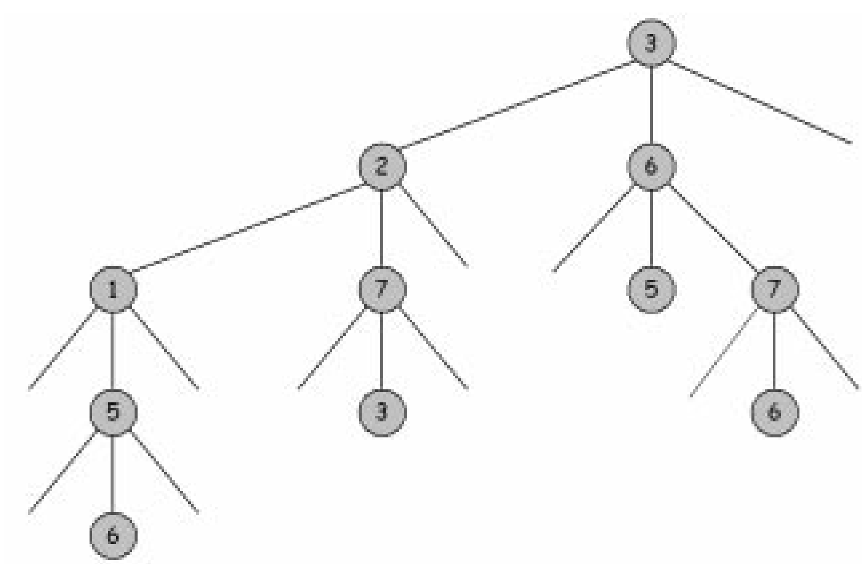
(9) Selection sort

(10) Shell sort

2. **Heaps. (5 points)**

   What is the minimum number of Item exchanges during a *delete the maximum* operation in a 15-element binary heap containing the keys 1–15.

3. **Tries. (6 points)**



   (a) List the words that appear in the existence TST (ternary search trie) above.

   (b) Insert the word 226 into the TST above, updating the picture above to reflect the insertion.

4. **Choosing the right algorithms and data structures. (8 points)**

Circle the best answer to each of the following questions.

(a) What is the primary reason to use a binomial queue instead of a binary heap?

i. Faster average-case delete max
ii. Faster insert
iii. Less space usage
iv. Faster join (a.k.a. merge)
v. Faster worst-case delete max

(b) What is the primary reason to use a randomized BST instead of a binary heap?

i. Faster average-case delete max
ii. Faster insert
iii. Less space usage
iv. Faster search
v. Faster worst-case delete max

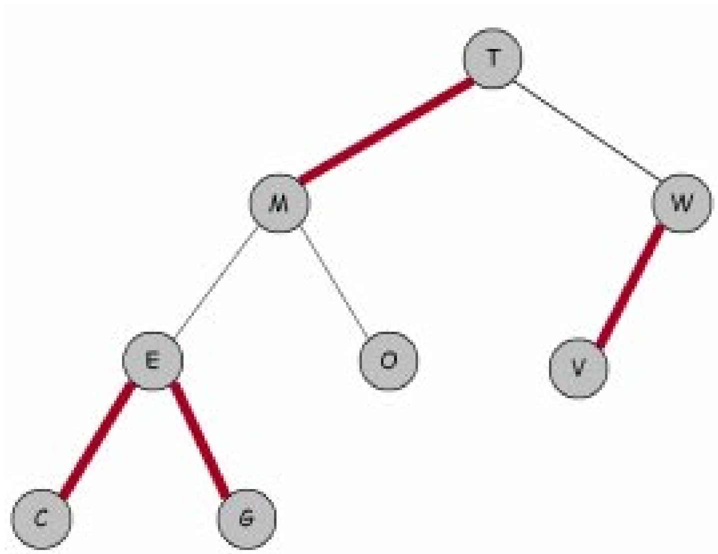
(c) What is the primary reason to use double probing instead of linear probing?

i. Faster search when the hash table is 50% full
ii. Faster insert when the hash table is 50% full
iii. Achieve same search times with less memory
iv. Easier to support delete
v. Easier to expand the hash table when it gets full

(d) What is the primary reason to use the Boyer-Moore right-to-left scan algorithm instead of the Knuth-Morris-Pratt algorithm?

i. Faster average-case search
ii. Faster worst-case search
iii. Less space usage
iv. Fewer lines of source code
v. No auxilliary buffer needed

5. **Red-black trees. (6 points)**



Insert the key Z, then the key A into the above red-black tree. Draw the resulting red-black tree in the space below.

6. **Programming assignments. (5 points)**

Given a list of N integers, the 3-sum problem is to determine whether there exists 3 integers in the list (not necessarily distinct) such that x + y + z = 0.

Suppose that you are executing the brute force algorithm below on a computer that executes 1 billion operations (addition, increment, or comparison) per second.

```
int brute(int a[], int N) {
   int i, j, k;
   for (i = 0; i < N; i++)
      for (j = 0; j < N; j++)
         for (k = 0; k < N; k++)
            if (a[i] + a[j] + a[k] == 0) return 1;
    return 0;
}
```

(a) Estimate how many seconds it will take (in the worst case) to solve a problem of size N = 1,000? Full credit if you are within 1% of the exact answer.

(b) Of size N = 10,000?

7. **Programming assignments. (10 points)**

Design an algorithm for the 3-sum problem that runs in $O(N^2 \log N)$ time and uses $O(N)$ memory. Give a high level description of your algorithm. Your answer will be graded on correctness, running time, memory usage, and the clarity and conciseness of your description.

Significant partial credit will be awarded if you use quadratic space. Bonus points will be awarded if you algorithm runs in $O(N^2)$ time and uses only $O(1)$ extra memory.

(a) What is the running time of your algorithm as a function of N? Explain briefly.

(b) What is the memory usage in bytes of your algorithm as a function of N, not including the input array `a[]`? Assume pointers and ints are 4 bytes each. Explain briefly.