Computer Science Foundation Exam

August 12, 2005

Computer Science

Section 1A

No Calculators!

KEY

Solutions and Grading Criteria

Name: _____

SSN: _____

Score: ¤ 50

In this section of the exam, there are four (4) problems. You must do all of them.

The weight of each problem in this section is indicated with the problem.

Partial credit cannot be given unless all work is shown and is readable.

Be complete, yet concise, and above all be neat.

1. [15 points – 5 pts each]

Answer each of the following "timing" questions concerning an algorithm of a particular order and a data set of a particular size. Assume that the run time is affected only by the size of the data set and not its composition and that **n** is an arbitrary integer. Show your work and place your final answer in the box.

(a) For an **O(n $\log_2$ n)** algorithm, one data set with **n = 32** takes **16** seconds. How long will the same algorithm take if the size of the data set is increased to **n = 256**?

Deduct 1 point for each simple math/algebra error. Deduct 3-5 points for improper technique depending on mistake.

20.48 sec

$$\frac{32\log_2 32}{16}=\frac{32(5)}{16}=10=\frac{256\log_2 256}{T}=10T=256(8)=T=\frac{2048}{10}=20.48\text{sec}$$

(b) For an **O(n^2)** algorithm, one data set with **n = 16** takes **24** seconds. What is the largest size data set that can be executed by this algorithm in **60** seconds?

Deduct 1 point for each simple math/algebra error. Deduct 3-5 points for improper technique depending on mistake.

n = 25

$$\frac{16^2}{24}=\frac{n^2}{60}\Rightarrow n^2=\frac{256\times 60}{24}\Rightarrow n^2=640\Rightarrow n=25$$

(c) Suppose you have two different algorithms that both correctly solve a particular problem. One of the algorithms is **O(n!)** and the other algorithm is **O(n^3)**. Which of the algorithms takes longer to execute when **n = 5**?

The $O(n^3)$ algorithm takes the longer time.

$5! = 1\times 2\times 3\times 4\times 5 = 120$

$5^3 = 5\times 5\times 5 = 125$

$125 > 120$ so the $O(n^3)$ algorithm takes longer when n = 5

Deduct 1 point for each simple math/algebra error. Deduct 3-5 points for improper conclusion depending on how/why they answered incorrectly.

For parts (a)-(c).
Deduct 1 point for each simple math/algebra error.
Deduct 3-5 points for improper technique depending on mistake (including use of wrong closed form).

2. [15 points – 5 pts each] Show all work a

(a) Given the following pseudocode seg terms of **n**.

```
x = 0;
for  i = n to 3*n do
   for j = 1 to n-2 do
       x = x + j;
```

$$\frac{2n^3 - 5n^2 + n + 2}{2}$$

$$\sum_{i=n}^{3n}\sum_{j=1}^{n-2} j = \sum_{i=n}^{3n}\frac{(n-2)(n-1)}{2} = \sum_{i=1}^{3n}\frac{n^2-3n+2}{2} - \sum_{i=1}^{n-1}\frac{n^2-3n+2}{2}$$

$$= \frac{3n(n^2-3n+2)}{2} - \frac{(n-1)(n^2-3n+2)}{2} = \frac{3n^3-9n^2+6n-n^3+4n^2-5n+2}{2}$$

$$= \frac{2n^3-5n^2+n+2}{2}$$

(b) Given the following pseudocode segment, determine the value of **x** when the **for** loops end in terms of **n**.

```
x = 0;
for i = 1 to 2*n do
    for j = 1 to n do
        if (j < i) then
            x = x + 1;
```

$$\frac{3n^2 - n}{2}$$

$$\sum_{i=1}^{2n}\sum_{j=1}^{n} 1 - \sum_{k=1}^{n} k = 2n^2 - \frac{n(n+1)}{2} = \frac{4n^2-n^2-n}{2} = \frac{3n^2-n}{2}$$

(c) Find the closed form for the following sequence.

S = 2 + 4 + 6 + 8 + … + 2(n-2) + 2(n-1)

$$n^2 - n$$

$$S = 2(1+2+3+4+\ldots+(n-2)+(n-1))$$

$$= 2\sum_{i=1}^{n-1} i = \frac{2(n-1)(n)}{2} = (n-1)(n) = n^2 - n$$

3. [10 points]

Show the results of each pass of the sorting algorithm of your choice on the following unsorted array. You may choose from (a) the bubble sort, (b) the selection sort, and (c) the insertion sort. Choose one (1) of these sorting techniques only and show the effect on the array after each pass of the sorting technique of your choice. Clearly indicate in the box provided the sorting algorithm you have selected. Assume the elements are to be sorted into ascending order. You may stop illustrating passes when the array becomes sorted.

The unsorted array

7	3	9	8	6	5	4	1	2

I have chosen the Bubble Sort – smallest to largest

The array as the sort progresses:

PASS	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
Initial	7	3	9	8	6	5	4	1	2
1	1	7	3	9	8	6	5	4	2
2	1	2	7	3	9	8	6	5	4
3	1	2	3	7	4	9	8	6	5
4	1	2	3	4	7	5	9	8	6
5	1	2	3	4	5	7	6	9	8
6	1	2	3	4	5	6	7	8	9

Deduct 3-4 points if they do not specify which sorting technique they utilized.

Deduct 4-5 points if they specified one technique but actually used another (mixed up the names of the sorts).

Deduct 1-2 points for improper positioning of values in the array after a pass has been made.

Often in CS1 when a question such as this appears on an exam, there will be several students who develop very "creative" sorts, so be sure that they are applying one of the specified sorting techniques.

I have chosen the Bubble Sort – largest to smallest

The array as the sort progresses:

PASS	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
Initial	7	3	9	8	6	5	4	1	2
1	3	7	8	6	5	4	1	2	9
2	3	7	6	5	4	1	2	8	9
3	3	6	5	4	1	2	7	8	9
4	3	5	4	1	2	6	7	8	9
5	3	4	1	2	5	6	7	8	9
6	3	1	2	4	5	6	7	8	9
7	1	2	3	4	5	6	7	8	9

I have chosen the Insertion Sort

The array as the sort progresses:

PASS	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
Initial	7	3	9	8	6	5	4	1	2
1	3	7	9	8	6	5	4	1	2
2	3	7	9	8	6	5	4	1	2
3	3	7	8	9	6	5	4	1	2
4	3	6	7	8	9	5	4	1	2
5	3	5	6	7	8	9	4	1	2
6	3	4	5	6	7	8	9	1	2
7	1	3	4	5	6	7	8	9	2
8	1	2	3	4	5	6	7	8	9

I have chosen the Selection Sort – smallest to largest

The array as the sort progresses:

PASS	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
Initial	7	3	9	8	6	5	4	1	2
1	1	3	9	8	6	5	4	7	2
2	1	2	9	8	6	5	4	7	3
3	1	2	3	8	6	5	4	7	9
4	1	2	3	4	6	5	8	7	9
5	1	2	3	4	5	6	8	7	9
6	1	2	3	4	5	6	8	7	9
7	1	2	3	4	5	6	7	8	9

I have chosen the Selection Sort – largest to smallest

The array as the sort progresses:

PASS	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
Initial	7	3	9	8	6	5	4	1	2
1	7	3	2	1	6	5	4	8	9
2	7	3	2	1	6	5	4	8	9
3	4	3	2	1	6	5	7	8	9
4	4	3	2	1	5	6	7	8	9
5	4	3	2	1	5	6	7	8	9
6	1	3	2	4	5	6	7	8	9
7	1	2	3	4	5	6	7	8	9

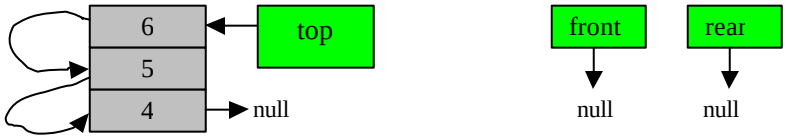
4. [10 points]

Given the sequence of operations shown below on their respective data structures, clearly show the final "state" (i.e., the contents) of each of the data structures. Assume that the data structures involved are dynamic (not arrays). **Draw your data structures carefully and label them appropriately. Points will be deducted if the data structures are not properly labled!**

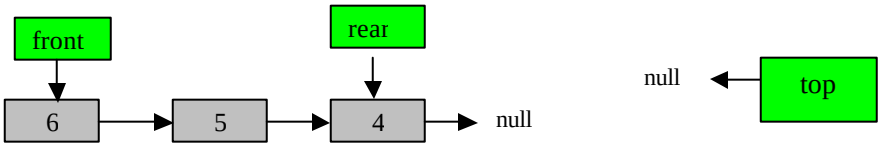
```
push(4)
push(5)
push(6)
enqueue(pop( ))
enqueue(pop( ))
enqueue(pop( ))
push(dequeue( ))
push(dequeue( ))
push(dequeue( ))
```

Deduct 4-5 points if the data structures are not properly labeled as to type and where the top, front, and rear are located.

I've had a question like this on CS1 exams before and sometimes the student will attempt to solve this problem with a single data structure rather than using push and pop on a stack and enqueue and dequeue on a queue.

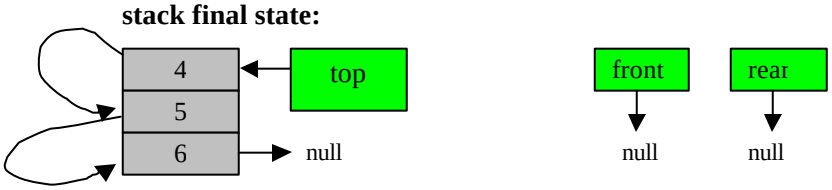


stack after first 3 push operations, queue empty at this point



queue after first 3 enqueue operations, stack is now empty after 3 pops

stack final state:



Code reversed contents of the stack using a queue.