Exam 2

Tuesday, March 31, 2020 5:07 PM

CS 248 **MIDTERM EXAM #2** **March 31st, 2020**

Name: N8 Swalley Score [75 possible]:

1. [16+2EC points] Matching.
Match each item on the left with its best match on the right. Please write the letters on the left side. There is no additional penalty for guessing. There are two extra matches for up to two points extra credit.

C	Awkwafina	(a)	Prolog
B	Base case	(b)	Solve directly, without recursion
A	Declarative language	(c)	Fleetfoot
I	Device driver	(d)	Unlimited memory illusion
H	Duplicated work	(e)	Portion of network communication
G	Dynamic programming	(f)	Java
O	Ethernet	(g)	Arrays for recursion
N	Functional language	(h)	Recursive Fibonacci pitfall
M	Grammar	(i)	Translates between the OS and devices
R	Imperative language	(j)	$T(n) = O(n) + 2T(n/2)$
Q	Kernel	(k)	$T(n) = O(n) + T(n-1)$
P	Lexical Analysis	(l)	Seaplane
L	Nick Jonas	(m)	Rules for language syntax
K	$O(n^2)$	(n)	LISP
J	$O(n \log n)$	(o)	Local broadcast network
F	Object-oriented language	(p)	Compiler phase
E	Packet	(q)	Operating system code
D	Virtual memory	(r)	Fortran

2. [15 points] Stacks and Queues.

(a) [10 points] The following code fragment runs without errors. What output does it produce? Show your work for partial credit.

```
Stack S=new StackLL();
Queue Q=new QueueLL();

S.push("J");
Q.enqueue("H");
Q.enqueue("U");
Q.enqueue("M");
Q.enqueue("A");
Q.enqueue("N");
Q.dequeue();

while(!S.isEmpty())
{
  System.out.print(S.getTop());
  Q.enqueue(S.pop());
}
while(!Q.isEmpty())
  { System.out.print(Q.dequeue()); }

S.push("A");
S.push("L");
S.push("I");
S.push("E");
Q.enqueue("N");
Q.enqueue(S.pop());

if(!S.isEmpty())
  { System.out.println(S.pop()); }
```

Queue: HUMANNJNA

Stack: E I L A J

Output: UMANJLIE

(b) [5 points] Evaluate this postfix expression:

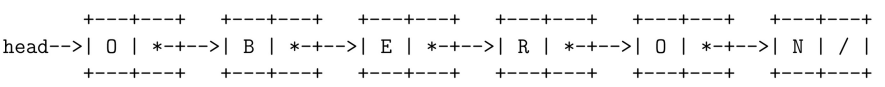
```
1 2 + 3 4 * 5 - +
```

1+2 + 3*4 -5

=10

3. [10 points] Linked Lists.
Write a function to get the last item in a linked list. Given a head pointer to a list, your function will find the last node of the list, and then return as its function value the item stored in that node.

Example: given the following list (please forgive the ASCII drawing):



your function should return "N". You may assume that the list has at least one node, or in other words, `head!=null`. Use the following declarations (assume they are embedded in a class).

```
class node
{
  public Object item;
  public node next;
}

public Object last(node head)
{
```

```
node p = head;
while (p.next != null)
{
  p = p.next
}

return p.item
```

4. [5 points] Extra Credit.
Solve the problem above using recursion for 5 points of extra credit. (So either you use recursion above, or you give a second version here that is recursive.)

5. [10 points] Recursion.
What is the output of the code fragment below? Also show the values of the `mouse()` function you need for full credit. (Remember that, for example, `'A'+3` is `'D'`.)

```
public static int mouse(int x)
{
  if(x==0)   return 2;
  if(x%4==0) return mouse(x-1)+3;
  else       return mouse(x-1)+4;
}
```

j(4) : 'A' + m(4) = 17

return m(4-1) + 3

→ m(3) = 14

```
  if(x==0)   return 2;
  if(x%4==0) return mouse(x-1)+3;
  else       return mouse(x-1)+4;
}

public static void jurgen(int x)
{
  System.out.print((char)('A'+mouse(x)));
}

...

jurgen(4); jurgen(3); jurgen(0); jurgen(2);
System.out.println("");
```

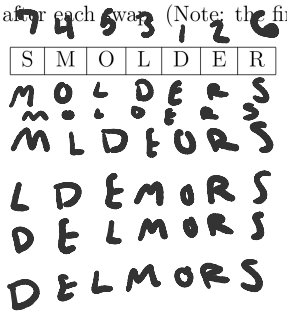
Output: 'R' 'O' 'L' 'K'

→ m(3) = 14
return m(3-1) + 4
→ m(2) = 10
return m(2-1) + 4
→ m(1) = 6 ✓
return m(1-1) + 4
return 6
m(0) = 2

6. [24 points] Sorting Algorithms.
Answer each of the following questions about sorting algorithms.
Please read the directions carefully!
In all cases, use the version of the algorithm we implemented in class.

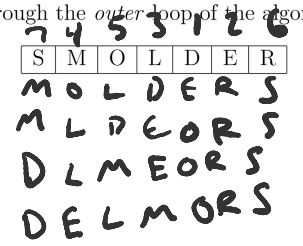
(a) Show how Selection Sort would process this same array. Show what the array looks like after each swap. (Note: the final result should be sorted.)

S	M	O	L	D	E	R



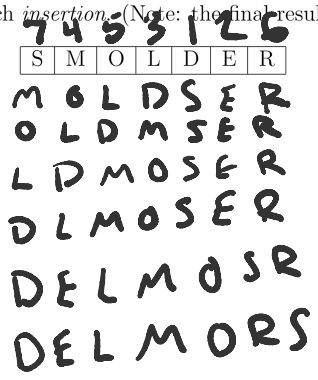
(b) Show how Bubble Sort would process the array below. Give the result after one pass through the *outer* loop of the algorithm.

S	M	O	L	D	E	R



(c) Show how Insertion Sort would process this array. Show what the array looks like after each *insertion*. (Note: the final result should be sorted.)

S	M	O	L	D	E	R



(d) Show how Shell Sort would process this array. Using a gap value of 2 (both offsets), show what the array looks like after the one pass.

S	M	O	L	D	E	R

D O R S

D E O L R M S

(e) Show what the array below would look like after executing the partition function from Quicksort exactly once. Assume the last array element ('R') is used as the pivot.

S	M	O	L	D	E	R

M S O L D E Big = ∅1

M S L O D E

D E L M O S → R DEL MORS

(f) Show what the following array portion looks like after executing the merge function (we called it `domerge()`) from the Mergesort algorithm. (Note that each half has already been sorted recursively, with midpoint 3.)

L	M	O	S	D	E	R

D E O R L M S

---

7. [5 points] Extra Credit.

What does the magic water do? Burns

How do they escape the ostriches? Run

Who flies at the end, never to return? That weird dragon dog thing from the neverending story

Name two fish pillows that appeared in videos (2). Angel & Dory