Name: SOLUTIONS

Student ID #:

PSO Section:

Notes:

❏ This is a closed book, closed notes exam. Any evidence of improper conduct will be dealt with strictly in accordance with university policies relating to cheating.

❏ All logarithms are in base 2 unless otherwise indicated.

❏ All answers must be clearly written with appropriate explanations. No partial credit will be awarded if appropriate steps are not outlined.

---

1. (2 x 5 = 10 points) Indicate whether the following are true or false with a one line justification (no points for missing or incorrect justification).
(GRADED BY Mr. Boon-Yeong Tan)

a) Java is more portable than C because it is object oriented.

FALSE, its because it is compiled to bytecode which is executed on the JVM.

b) Java programs are compiled into machine language using the java compiler.

FALSE, they are compiled into byte code.

c) Object oriented programs have more errors because they are more complicated.

FALSE, they have fewer errors due to encapsulation.

d) Java programs can only be executed from within browsers.

FALSE, they can be executed as stand-alone applications as well.

e) A new copy of a class variable is created every time a new instance of the class is created.

FALSE, class variables are shared by all instances of a class.

2. (3 x 5 = 15 points) Derive the best big-Oh notation for the following:
(GRADED BY Mr. Boon-Yeong Tan)

a. n + 1/n

O(n);

Since n + 1/n < 2n for n > 2.

b. log (n) + log (sqrt(n))

O(log(n))

This follows from the fact that log(sqrt(n)) = 0.5 log(n).

c. 1 + 2 + ... + log(n)

$O(\log^2 n)$

This follows from the fact that the above series sums to log(n) * (log(n) + 1) / 2

3. (3 x 5 = 15 points) Derive the best big-Omega notation for the following:
(GRADED BY Ms. Wei Zhou)

a. n - 1/n

$\Omega(n)$

This follows from the fact that n - 1/n > 0.5 n for n > 2

b. 1 + 2 + ... + (n-1)

$\Omega(n^2)$

This follows from the fact that the series sums to (n-1)n / 2.

c. log n + log $(2^{(n+1)})$

$\Omega(n)$

This follows from the fact that the expression is identical to log n + (n + 1)

4. (5 + 5 = 10 points) What is the runtime (best big-Oh bound) for the following code:
(GRADED BY Ananth Grama)
(a)

```
for i = 2 to n - 1
     for j = i-2 to i
          a[i] += a[j]
     end for
end for
```

```
O(n)

This follows from the fact that the inner loop executes only
thrice for each of the (n-2) outer iterations.
```

(b)

```
for i = 1 to n
     if (i < 1) then
          for j = 1 to i
               sum += a[j]
          end for
     end if
end for
```

```
O(n)

This follows from the fact that the inner loop is never executed.
```

5. (10 points) Using induction, show that
(GRADED BY Ms. Wei Zhou)

$1 + a + a^2 + a^3 + ... + a^n = (1 - a^{(n + 1)}) / (1 - a)$

Base case:

$n = 0$

LHS: 1
RHS: $(1-a) / (1 - a) = 1$

Therefore hypothesis holds for $n = 0$

(you could also have used $n = 1$ as the base case and no points would be deducted)

Assumption:

The hypothesis holds for $n = r$; i.e.

$1 + a + a^2 + a^3 + ... + a^r = (1 - a^{(r + 1)}) / (1 - a)$ .... (1)

Induction step:

For $n = r + 1$

LHS: $1 + a + a^2 + a^3 + ... + a^r + a^{r+1}$
$= (1 - a^{(r + 1)}) / (1 - a) + a^{r+1}$ (follows from equation (1))
$= (1 - a^{(r + 1)} + a^{(r + 1)} - a^{(r + 2)}) / (1 - a)$
$= (1 - a^{(r + 1)}) / (1 - a)$ (which is equal to the RHS for n = r+1)

Therefore hypothesis holds for $n = r + 1$

Therefore by induction, the hypothesis holds for all $n > 0$.

6. (20 points) Implement the stack ADT: `push(element), pop(), and size()` using the Queue ADT: `enqueue(element), dequeue(), and size()`. Derive complexities for each method in your stack ADT.
(GRADED BY Mr. Sarat Babu Kamisetty)

```
class stack {

     Queue q, temp;

     Stack() {
          q = new Queue();
          temp = new Queue();
     }

     int size() {
          return (q.size());
     }

     void push(Object o) {
          q.enqueue (o);
     }

     Object pop() {
          int i, num_items;
          Object ret;
          num_items = q.size();
          for (i = 0; i < num_items - 1; i++)
               temp.enqueue(q.dequeue());
          ret = q.dequeue();
          for (i = 0; i < num_items-1; i++)
               q.enqueue(temp.dequeue());
          return ret;
     }
}

Complexities of size and push are O(1)
Complexity of pop is O(n)


NOTE: you could also have made pop identical to dequeue and used
two queues to implement push.
```

7. (10 points) Pictorially illustrate an initially empty ordered sequence S after each of the following operations:
(GRADED BY Mr. Ameya Ashok Limaye)

```
S.insertFirst(c)

c
```

```
S.insertLast(a)

c a
```

```
S.insertAtRank(1, b)

c b a
```

```
S.remove(S.after(S.first()))

c a
```

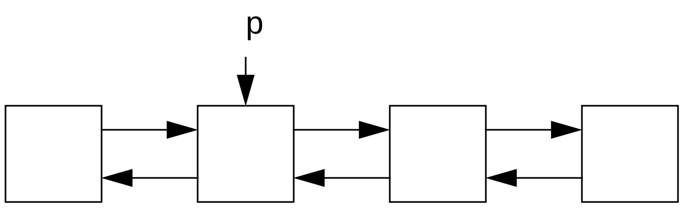
```
S.insertBefore(S.last(), d)

c d a
```

8. (10 points) Write a pseudocode for the `insertAfter(p,e)` method using a doubly linked list. Illustrate the operation pictorially as well.
(GRADED BY Ananth Grama)

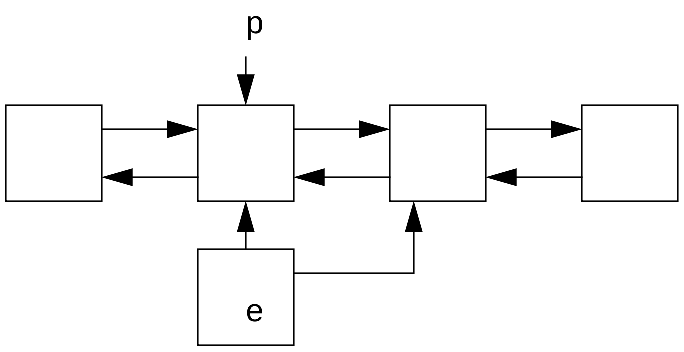
```
insertAfter(p, e) {
        Node temp = new Node (p, p.next, e);     // stmt 1
        p.next.prev = temp;                      // stmt 2
        p.next = temp;                           // stmt 3
}
```

Pictures:

Before stmt 1



After stmt 1



After stmts 2, and 3