University of Nevada, Las Vegas Computer Science 477/677 Spring 2020

Answers to Examination February 27, 2020

The entire examination is 250 points.

1. True or False. [5 points each]

   (a) T Graham Scan takes $O(n \log n)$ time to find the convex hull of n points in a plane.

   (b) T The traditional implementation of selection sort takes quadratic time.

   (c) T If pivots are picked at random, the expected number of comparisons needed by quicksort is $O(n \log n)$, regardless of the initial array.

2. Fill in the blanks. [5 points each blank.]

   (a) Give three kinds of priority queue.
   stack
   queue
   heap

   (b) Name a well-known divide-and-conquer searchng algorithm. **binary search**

   (c) Any comparison-based sorting algorithm takes at least $\log_2 n!$ comparisons to sort n items.

3. Solve the recurrences. Give asymptotic answers in terms of n, using either O, Ω, or Θ, whichever is most appropriate. [10 points each.]

   (a) $F(n) = 4F(n/2) + n$

   $F(n) = \Theta(n^2)$

   (b) $F(n) = 4F(n/2) + n^3$

   $F(n) = \Theta(n^3)$

   (c) $F(n) \leq F(n/2) + F(n/4) + 4n$

   $F(n) = O(n)$

   (d) $F(n) \leq F(n-1) + 2F(n-2)$

   $F(n) = O(2^n)$

   (e) $F(n) = F(n/5) + 2F(2n/5) + F(4n/5) + n^2$

   $F(n) = \Theta(n^2 \log n)$

   (f) $F(n) \geq F(n-2) + n$

   $F(n) = \Omega(n^2)$

   (g) $F(n) = F(\log n) + 1$

   $F(n) = \Theta(\log^* n)$

4. Give the asymptotic time complexity of each of these code fragments, in terms of n. Use O, Ω, or Θ, whichever is most appropriate. [10 points each.]

(a)
```
for(int i = 1; i < n; i++)
 for(int j = i; j < n-i; j++)
  cout << "Hello" << endl;
```
$\Theta(n^2)$

(b)
```
for(int i = 1; i < n; i++)
 for(int j = i; j < n; j = 2*j)
  cout << "Hello" << endl;
```
$\Theta(n)$

(c)
```
for(int i = n; i > 0; i=i/2)
 for(int j = i; j < n; j = j++)
  cout << "Hello" << endl;
```
$\Theta(n \log n)$

(d)
```
for(int i = 1; i < n; i = 2*i)
 for(int j = 1; j < n; j = 2*j)
  cout << "Hello" << endl;
```
$\Theta(\log^2 n)$

(e)
```
for(int i = 2; i < n; i = i*i)
 cout << "Hello" << endl;
```
$\Theta(\log \log n)$

5. [20 points] Consider the following procedure:

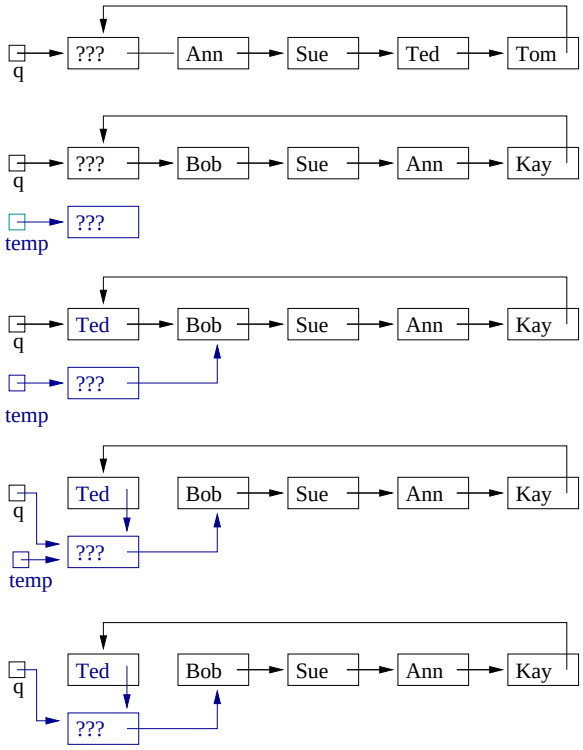
```
void augustine(int n)
 {
  if(n > 0)
   for (int i = 1; i < n; i++)
    cout << "George, did you chop down the cherry tree?" << endl;
  if(n > 0)
   {
    augustine(n/2);
    augustine(n/2);
   }
 }
```

Consider the question of how many lines of output the execution of `augustine(n)` would produce. Write down an appropriate recurrence for this question, and give an asymptotic solution in terms of `n`, using either O, Ω, or Θ, whichever is most appropriate.

$A(n) = 2A(n/2) + n$

$A(n) = \Theta(n \log n)$

6. [20 points] Illustrate a queue implemented as a circular linked list with a dummy node. The contents of the queue, from front to rear, should be the four items Bob, Sue, Ann, Kay. Show the steps of inserting the item Ted into the queue.



7. [30 points] Walk through polyphase mergesort, where the input file is as given below.

MDYCOSEZVQWBANLH

MCOSVBNH
DYEZQWAL

DMYBNQW
CEOSVZAHL

CDEMOSVYZ
ABHLNQW

ABCDEHLMNOSQVWYZ

8. [20 points] The following code implements the "primitive" algorithm for multiplication, where `x` and `y` are integers, and `y` is non-negative.

   What is the loop invariant of this code? Indicate the four places in the code where the loop invariant holds.

   Loop Invariant: `x*y = result + a*b`

```
int product(int x,int y) // returns x*y
 {
  int a = x;
  int b = y;
  int result = 0;
    LOOP INVARIANT HOLDS HERE
  while (b > 0)
   {
      LOOP INVARIANT HOLDS HERE
    if(b%2) result = result + x;
    a = a+a;
    b = b/2;
      LOOP INVARIANT HOLDS HERE
   }
    LOOP INVARIANT HOLDS HERE
  return result;
 }
```