University of Nevada, Las Vegas Computer Science 477/677 Fall 2023

Study for Examination February 12, 2025

1. Fill in the blanks.

   (a) Any comparison-based sorting algorithm on a file of size n executes _____ comparisons. (Use Ω notation.)

   (b) Name two well-known divide-and-conquer sorting algorithms.

   _____

   _____

2. Fill in each blank. Write Θ if that is correct; otherwise write O or Ω, whichever is correct. Recall that log means $\log_2$.

   (a) $\log n^2 =$ _____$(\log n^3)$

   (b) $\log(n!) =$ _____$(n \log n)$

   (c) $\sum_{i=0}^{n-1} i^k =$ _____(n^k)

   (d) $2^{\log^2 n} =$ _____(n^n)

   (e) $\log n =$ _____$(\ln n)$

3. Fill in the blanks.

4. True or False.

   (i) _____ A good programmer would never use an unordered list as a search structure.

   (ii) _____ If $\log F(n) = \Theta(\log G(n))$, then $F(n)$ must be $\Theta(G(n))$.

5. Fill in the blanks.

   (iii) The asymptotic time complexity to find an item in an ordered array of length n is _____, using the divide and conquer algorithm _____ _____.

   (iv) In a priority queue, each item in the structure represents an _____ _____.

   (v) Three kinds of priority queues we've mentioned in class are _____, _____, and _____.

   (vi) The height of a binary tree with 20 leaves must be at least _____. Exact answer please: no partial credit.

   (vii) "pop" and "push" are operators of _____.

   (viii) "fetch" and "store" are operators of _____.

(ix) "insert" and "deletmin" are operators of _____.

6. Find the asymptotic time complexity of each of these code fragments in terms of n, using Θ notation.

7. A stack of integers could be implemented in C++ as a linked list as follows.

```
struct stack
 {
  int item;
  stack*link;
 };
```

Finish writing the code for the operators push, pop, and empty, below.

```
void push(stack*&s,int newitem)




bool empty(stack*s)



int pop(stack*&s)
 {
  assert(          // what do you need to assert?
```

8. Let $W_1, W_2, \ldots$ be the Wonderful numbers, an increasing sequence. The Wonderful numbers have the property that $W_i = 2W_{i-1} + 2W_{i-2}$ for all $i > 2$, and the first few Wonderful numbers are 1, 2, 6, 16, 44, ... Find a constant K such that $W_n = \Theta(K^n)$. Show your work.

9. The following portion of C++ code contains an array implementation of `queue`. Fill in the missing code for the operators "`enqueue`" and "`empty`."

```
struct queue
 {
  int A[N]; // N is a constant large enough to prevent false overflow.
  int rear = 0;
  int front = 0; // initially the queue is empty
 };

void enqueue(queue&q,int newitem) // inserts newitem into q




bool empty(queue&q) // returns true if q is empty, false otherwise



int dequeue(queue&q) // returns an item from q and deletes that item
 {
  asssert(not empty(q));
  int rslt = q.A[q.front];
  q.front++;
  return rslt;
 }
```

10. Exactly one of the following is true for a binary tree with n nodes. Which one?

    The height of the tree is $O(\log n)$.
    The height of the tree is $\Theta(\log n)$.
    The height of the tree is $\Omega(\log n)$.

11. The items B, Q, T, M, A, X, L, are inserted into a binary search tree in that order.

    (a) List the items of the tree in preorder.

    (b) List the items of the tree in inorder.

    (c) List the items of the tree in postorder.

    (d) List the items of the tree in level order.

12. What follows is part of a C++ implementation of binary tree of integer. (It runs, I checked.) Fill in the missing code for the recursive functions `insert`, `find`, and `lcation`. The function `lcation` should be called `find`, but that would cause a redundancy conflict.

```
struct treenode;
typedef treenode*tree;
struct treenode
 {
  int item;
  tree left;
  tree right;
 };

tree mainroot; // the root of the binary search tree

void insert(tree&root,int n)
  // inserts n into the bineary search tree.  Duplicates are not inserted.
 {
  if(root == NULL)
   {
    root = new treenode;
    root->item = n;
   }
    // 3 lines missing









 }

bool find(tree root, int n)
  // returns true if some node contains n, false otherwise.
 {
  // 4 lines missing









 }
```

```
tree lcation(tree root, int n)
  // returns pnter to the node which contains n, if any; NULL otherwise.
 {
  // 4 lines missing









 }

void getdata()
  // insert data into binary search tree
 {
  // Whatever code is needed for the application
 }

int main()
 {
  getdata();
  // Whatever code is needed for the application
  return 1;
 }
```

13. A min-heap implemented as a binary tree which is, in turn, implemented as an array, is shown in the first row of the array below. In the subsequent rows, show the evolution of the heap when "B" is inserted.

9	A	H	C	M	N	F	G	P	R	K

14. Rewrite the infix expression $a - -b * c * (d + -e) * f$ in prefix notation, and then in postfix notation.

15. what well-known algorithm does the procedure `mystery1` implement? What well-known algorithm does `mystery2` implement? What well-known algorithm does `mystery3` implement? Assume that A has been declared as int A[N].

```
void swap(int x, int y)
 {
  int z = x;
  x = y;
  y = z;
 }

void mystery1()
 {
  for(int i = 0; i < N; i++)
   for(int j = i+1; j < N; j++)
    if(A[j] < A[i]) swap(A[i],A[j]);
 }

void mystery2()
 {
  bool finshd = false;
  while(not finshd)
   {
    finshd = true;
    for(int i = 0; i < N-1; i++)
     if(A[i+1] < A[i])
      {
       finshd = false;
       swap(A[i],A[i+1]);
      }
   }
 }
```

```
void mystery3()
 {
  for(int i = 1; i < N; i++)
   {
    bool fnshd = false;
    for(int j = i; j > 0 and not finshd; j--)
     if(A[j] < A[j-1]) swap(A[j],A[j-1]);
     else fnshd = true;
   }
 }
```

16. Show a circular queue with dummy node, with items B, M, Q, R, in that order from front to rear. Then show how the queue changes when you insert H, and then show how the queue changes when you execute dequeue.

Ass 1:

1. Write a C++ function which determines whether a given integer, which is at least 2, is prime.

```
bool prime(int n)
 // input condition: n >= 2
 {

 }
```

Ass 2

1. Write either O, Ω or Θ in each blank. Write Θ if that is correct, otherwise write O or Ω.

   (a) $n - 100 =$ _____ $(n - 200)$

   (b) $n^{1/2} =$ _____ $(n^{2/3})$

   (c) $100n + \log n =$ _____ $(n + \log^2 n)$

   (d) $n \log n =$ _____ $(10n + \log(10n))$

   (e) $\log(2n) =$ _____ $(\log(3n))$

   (f) $10 \log n =$ _____ $(\log(n^2))$

   (g) $n^{1.01} =$ _____ $(n \log^2 n)$

   (h) $n^2 / \log n =$ _____ $(n \log^2 n)$

(i) $n^{0.1} =$ ------ $(\log^2 n)$

(j) $(\log n)^{\log n} =$ ------ $(n/\log n)$

(k) $\sqrt{n} =$ ------ $(\log^3 n)$

(l) $n^{1/2} =$ ------ $(5^{\log_2 n})$

(m) $n2^n =$ ------ (3^n)

(n) $2^n =$ ------ (2^{n+1})

(o) $n! =$ ------ (2^n)

(p) $\log_2 n^{\log_2 n} =$ ------ $(2^{(\log_2 n)^2})$

(q) $\sum_{i=1}^{n} i^k =$ ------ (n^{k+1})

2. Let $F_1, F_2, F_3 \ldots$ be the Fibonacci numbers Recall that $F_i + F_{i+1} = F_{i+2}$. The first few Fibonacci numbers are 1, 1, 2, 3, 5, 8, ...

   Find the smallest constant C such that $F_n = O(C^n)$.

3. Consider the following C++ program.

```
void process(int n)
 {
  if(n > 1) process(n/2);
  cout << n%2;
 }

int main()
 {
  int n;
  cout << "Enter a positive integer: ";
  cin >> n;
```

```
  assert(n > 0);
  process(n);
  cout << endl;
  return 1;
}
```

The output of `process(n)` is a string of bits. What does this bitstring represent?

4. The C++ code below implements a function, "mystery4." What does it compute?

```
float mystery4(float x, int k)
 {
  if (k == 0) return 1.0;
  else if(x == 0.0) return 0.0;
  else if (k < 0) return 1/mystery(x,-k);
  else if (k%2) return x*mystery(x,k-1);
  else return mystery(x*x,k/2);
 }
```

5. The C++ code below implements a function. What does that function compute?

```
int gcd(int n, int m)
 {
  if(n < 0) return gcd(-n,m);
  else if(m < 0) return gcd(n,-m);
  else if(n < m) return gcd(m,n);
  else if(m > 0) return gcd(m,n%m);
  else return n;
 }
```

Ass 3

1. Give the names of three kinds of priority queue.

   _____ _____ _____

2. The answer to each of these questions is either **bubbleup** or **bubbledown**.

   If a heap is implemented a binary tree:

   (a) Insertion into the heap requires the use of _____.

   (b) Deletmax (or deletemin, as the case may be) reuquires use of _____.

3. The following is C++ code for which quadratic time sorting algorithm?

   _____

```
int x[N];

void swap(int x, int y)
 {
  int temp = x;
  x = y;
  y = temp;
 }

void sort()
 {
  for(int i = 0; i < N; i++)
   for(int j = i+1; j < N; j++)
    if(x[j] < x[i]) swap(x[i],x[j]);
 }
```

4. The following is C++ code for a function that computes a floating point number to the power of a positive integer. Find a loop invariant which can be used to prove correctness of the function.

```
float power(float x, int n) // input condition: n > 0
 {
  assert(n > 0);
  float z = 1.0;
  float y = x;
  int m = n;
  while(m > 0)
   {
    if(m%2) z = y*z;
    y = y*y;
    m = m/2;
   }
  return z;
 }
```

5. We can experimentally analyze the time complexity of code by using a counter. In the following example, we count the number of iterations of the loop.

```
   cout << "Enter a positive integer" << endl;
   int n;
   cin >> n;
   int kount = 0;
   for(int i = 0; i < n; i++)
    kount++;
   cout << "kount = " << kount << endl;
```

Of course, the output will be n. We can say that the time complexity of the code is $\Theta(n)$.

For each of the code fragments below, find the asymptotic time complexity in terms of n. Assume that the value of n is given.

(a)
```
for(float x = n; x > 1; x = x/2)
```

(b)
```
for(int i = 0; i*i < n; i++)
```

(c)
```
for(float x = n; x > 2.0; x = sqrt(x))
```

(d)
```
for(int i = 1; i < n; i=2*i)
```

(e)
```
for(int i = n; i > 1; i=i/2)
```

(f)
```
for(int i = 1; i < n; i++)
 for(int j = 1; j < i; j = j*2)
```

(g)
```
for(int i = 1; i < n; i++)
 for(int j = i; j < n; j = j*2)
```

(h)
```
for(int i = n; i > 1; i--)
 for(int j = i; j > 1; j = j/2)
```

(i)
```
for(int i = n; i > 1; i--)
 for(int j = n; j > i; j = j/2)
```

(j)
```
for(int i = 1; i*i < n; i++)
```

6. Walk through mergesort with the following input:
   XYRGEPMS

7. Walk through polyphase mergesort with the following input:
   MAHKWMRVULFB

8. Create a treap with the following itmes, each with the priority shown in the table. Use min-heap order.

item	priority
V	3
A	5
S	6
Q	4
N	7
K	2