

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

Study for Examination February 7, 2024

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) $\log n^2 = \text{_____}(\log n^3)$

(b) $\log(n!) = \text{_____}(n \log n)$

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

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

(e) $\log n = \text{_____}(\ln n)$

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 10 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 -----

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

(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))`

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, \dots be the Wonderful numbers. Similarly to the Fibonacci numbers, each Wonderful number is twice the sum of the previous two Wonderful numbers. 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
{
    assert(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 `location`. The function `location` 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 binary 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 location(tree root, int n)
    // returns pointer 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? 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]);
            }
    }
}
```

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