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

Answers to Study Guide for Examination February 12, 2025

1. Fill in the blanks.

   (a) Any comparison-based sorting algorithm on a file of size n executes $\Omega(n \log n)$ comparisons. (Use Ω notation.) Alternative answer: $\Omega(\log n!)$.

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

   quicksort **mergesort**

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) $\log n^2 = O(\log n^3)$

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

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

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

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

3. True or False.

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

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

4. Fill in the blanks.

   (iii) The asymptotic time complexity to find an item in an ordered array of length n is $O(\log n)$, using the divide and conquer algorithm **binary search**.

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

   (v) Three kinds of priority queues we've mentioned in class are **stack**, **queue**, and **heap**.

   (vi) The height of a binary tree with 20 leaves must be at least **5**. Exact answer please: no partial credit.

   (vii) "pop" and "push" are operators of **stack**.

   (viii) "fetch" and "store" are operators of **array**.

   (ix) "insert" and "deletmin" are operators of **minheap**.

5. 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)
 {
  stack*tmp = new stack;
  tmp->link = tmp;
  s = tmp;
 }

bool empty(stack*s)
 {
  return s == NULL;
 }


int pop(stack*&s)
 {
  assert(not empty(s));
  s = s->link; // memory leak here
 }
```

6. 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.

   We can assume that $W_i = K^i$.

$$\begin{aligned} W_i &= 2W_{i-1} + 2W_{i-2} \\ K^i &= 2K^{i-1} + 2K^{i-2} \\ K^2 &= 2K + 2 \\ K^2 - 2K - 2 &= 0 \\ K &= \frac{2 \pm \sqrt{12}}{4} \\ &= \frac{1 \pm \sqrt{3}}{2} \end{aligned}$$

   Since the sequence is increasing, $K > 0$. Thus $K = \dfrac{1 + \sqrt{3}}{2}$.

7. 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
 {
  q.rear++;
  q.A[q.rear] = newitem;
 }

bool empty(queue&q) // returns true if q is empty, false otherwise
 {
  return q.rear == q.front;
 }

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;
 }
```

8. 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 $\Omega(\log n)$.

   The third one, $\Omega(\log n)$.

9. 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. BAQMLTX

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

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

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

10. 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;
   }
  else if(n < root->item)
   insert(root->left,n);
  else if(n > root->item)
   insert(root->right,n);
 }

bool find(tree root, int n)
  // returns true if some node contains n, false otherwise.
 {
  if(root)
   if(root->item == n) return true;
   else if(n < root->item) return find(root->left,n);
   else return find(root->right,n);
  else return false;
 }

tree lcation(tree root, int n)
  // returns pointer to the node which contains n, if any; NULL otherwise.
 {
  if(root)
   if(root->item == n) return root;
```

```
     else if(n < root->item) return lcation(root->left,n);
     else return lcation(root->right,n);
    else return NULL;
   }

  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;
   }
```

11. 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
10
10
10

A	H	C	M	N	F	G	P	R	K
A	H	C	M	N	F	G	P	R	B
A	H	C	M	B	F	G	P	R	N
A	B	C	M	H	F	G	P	R	N

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

$-a * * * \sim bc * d \sim ef$

$ab \sim c * de \sim + * f * -$

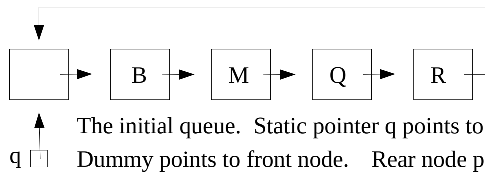
13. What well-known algorithm does the procedure `mystery1` implement? What well-known algorithm does `mystery2` implement? What well-known algorithm does `mystery3` implement? Assume that A has been declared as int A[N].

mystery1 is selection sort mystery2 is bubblesort mystery3 is insertion sord

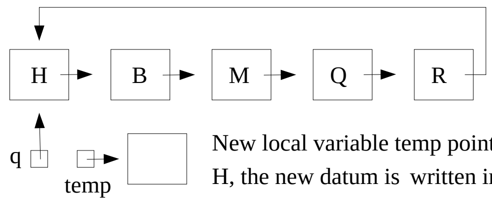
```
void swap(int x, int y)
 {
  int z = x;
  x = y;
  y = z;
 }
```

```
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;
   }
 }
```

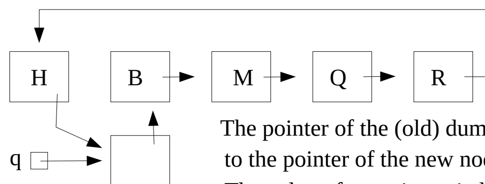
14. Show a circular queue with dummy node, with items B, M, Q, R, in that order from front to rear. Then show how the queue changes when you insert H, and then show how the queue changes when you execute dequeue.



The initial queue. Static pointer q points to the dummy node.
Dummy points to front node. Rear node points to dummy.
All nodes are private; q is the only publically visible part of the queue.



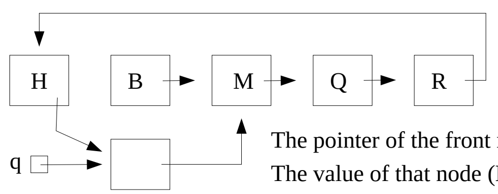
New local variable temp points to a new node.
H, the new datum is written into the dummy node.



The pointer of the (old) dummy node is copied to the pointer of the new node.
The value of temp is copied to the pointer q

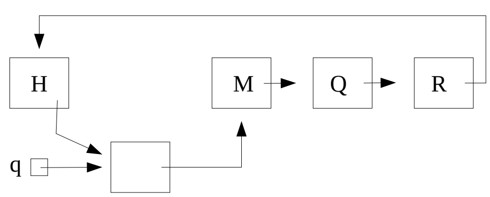
The new node becomes the dummy node, and the old dummy is the rear node. temp is deallocated. Static q is still the only public part of the structure..

Now execute dequeue.



The pointer of the front node is copied to q.
The value of that node (B in this example) is returned.

If memory space is a problem, deallocate the old front node.



---

15. 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
 int sq = sqrt(n);
 {
  bool rslt = true;
  for(int m = 2; m <= sq and rslt; m++)
   rslt = n%m
  return rslt;
 }
```

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

    (a) $n - 100 = \Theta\,(n - 200)$

    (b) $n^{1/2} = O\,(n^{2/3})$

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

    (d) $n \log n = \Omega\,(10n + \log(10n))$

    (e) $\log(2n) = \Theta\,(\log(3n))$

    (f) $10 \log n = \Theta\,(\log(n^2))$

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

    (h) $n^2/\log n = \Omega\,(n \log^2 n)$

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

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

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

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

    (m) $n2^n = O\,(3^n)$

    (n) $2^n = \Theta\,(2^{n+1})$

    (o) $n! = \Omega\,(2^n)$

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

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

17. 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 $F_n = O(C^n)$.

    We can assume that $F_n = C^n$. We obtain the quadratic equation $C^2 = C + 1$. Applying the quadratic formula, since $C > 0$, $C = \dfrac{1+\sqrt{5}}{2}$

18. 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?

    The binary numeral for n.

19. 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);
 }
```

    x^k

20. 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;
 }
```

The greatest common divisor of n and m.

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

    stack queue heap

22. The answer to each of these questions is either **bubbleup** or **bubbledown**. A heap is implemented a binary tree:

    (a) Insertion into the heap requires the use of **bubbleup**.

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

23. 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;
 }
```

$m \geq 0$ and $x^n = y^m * z$

24. 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)`

    $\Theta(\log n)$

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

    $\Theta(\sqrt{n})$

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

$\Theta(\log \log n)$

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

$\Theta(\log n)$

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

$\Theta(\log n)$

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

$\Theta(n \log n)$

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

$\Theta(n)$

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

$\Theta(n \log n)$

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

$\Theta(n)$

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

$\Theta(\log \log n)$

25. Walk through mergesort with the following input:

XYRGEPMS

XYRG EPMS

XY RG EP MS

XY GR EP MS

GRXY EMPS

EGMPRSXY

26. Walk through polyphase mergesort with the following input:

MAHKWMRVULFB

MMRVLB

AHKWUF

AHKMMRVWBF

LU

AHKLMMRUVW

BF

ABHKFLMMRUVW

27. 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