University of Nevada, Las Vegas Computer Science 477/677 Fall 2025

Answers to Examination September 18, 2025

The entire examination is 215 points.

1. Fill in the blanks.

   (a) [5 points] No good programmer would ever use an unsorted list as a search structure. **F** (True or False.)

   (b) [5 points] Heapsort is a fast version of **selection sort.**

   (c) [5 points] Treesort is a fast version of **insertion sort.**

   (d) [5 points] Shell sort is inspired by **bubblesort** but is faster.

   (e) [5 points] For any sorting algorithm where all branch points in the code are comparisons, the number of comparisons while sorting n items is $\Omega(n \log n)$.

   (f) [5 points] Name two different divide-and-conquer sorting algorithms.
   quicksort
   mergesort

   (g) [5 points] The items of a priority queue represent **unfulfilled obligations**

   (h) [5 points] Name three kinds of priority queues: **stack queue heap**

   (i) The worst case asymptotic time coplexity of sorting an array of length n is:

      i. [5 points] $O(n^2)$ using bubblesort.

      ii. [5 points] $O(n^2)$ using quicksort.

      iii. [5 points] $O(n \log n)$ using heapsort.

2. Write the asymptotic time complexity, in terms of n, of each of these code fragments.

   (a) [5 points]

```
for(int i = 0; i < n; i++)
 for(int j = 0; j < i; j++)
  cout << "Hello world" << end;
```

   $O(n^2)$

   (b) [5 points]

```
for(int i = 0; i < n; i++)
 for(int j = 1; j < i; j = 2*j)
  cout << "Hello world" << end;
```

   $O(n \log n)$

   (c) [5 points]

```
for(int i = 0; i < n; i++)
 for(int j = i; j < n; j = 2*j)
  cout << "Hello world" << end;
```

$O(n)$

(d) [5 points]

```
for(int i = 0; i < n; i++)
 for(int j = 0; j*j < i; j++)
  cout << "Hello world" << end;
```

$O(n\sqrt{n}) = O(n^{\frac{3}{2}})$

3. [10 points] Given a C++ implementation of queue of integer, finish the function dequeue.

```
struct queuenode;
typedef queuenode*queue;
const int N = 20;

struct queuenode
 {
  int item[N];
  int rear;
  int front;
 };

int dequeue(queue&q)
 // input condition: q.rear > q.front
 {
  int rslt = q.item[q.front];
  q.front++;
  return rslt;
 }
```

4. [10 points] Simplify each of these expressions.

(a) $\log 4 = 2$

(b) $\log_4 8 = \frac{3}{2}$

5. The following code computes the product of its two parameters. Write a loop invariant of the loop.

```
int product(float x, int n)
 {
  // input condition: n is non-negative
  float y = x;
  float z = 0.0;
  int m = n;
  while(m > 0) Error Fixed
   {
    if(m%2) z = z+y;
    y = 2*y;
    m = m/2;
   }
  return z;
 }
```

$x * n = y * m + z$

6. [20 points] Walk through the steps of heapsort, sorting an array whose items are X,B,L,F,U,N in that order. For ease of grading, use the matrix below. The number of rows in the matrix might not be exactly what you need. If not, leave rows blank or add extra rows.

X	B	L	F	U	N
X	B	N	F	U	L
X	U	N	F	B	L
L	U	N	F	B	**X**
U	L	N	F	B	**X**
B	L	N	F	**U**	**X**
N	L	B	F	**U**	**X**
F	L	B	**N**	**U**	**X**
L	F	B	**N**	**U**	**X**
B	F	**L**	**N**	**U**	**X**
F	B	**L**	**N**	**U**	**X**
B	**F**	**L**	**N**	**U**	**X**

7. [20 points] What follows is a C++ implementation of binary search tree with integer data.

```
struct treenode;
typedef treenode*tree;
struct treenode
{
 int item;
 tree lft = 0;
 tree rght = 0;
}
```

Write C++ code for each of these functions.

8. [10 points]

```
void insert(tree&root,int newitem)
 {
  if(root == 0)
   {
    root = new treenode;
    root->item = newitem;
   }
  else if(newitem < root->item)
   insert(root->lft,newitem)
  else
   insert(root->rgt,newitem)
 }
```

9. [10 points]

```
void inorderwrite(tree root)
 {
  if(root)
   {
    inorderwrite(root->lft);
    cout << root->item << endl;
    inorderwrite(root->rgt);
   }
 }
```

10. [10 points] Let F be an increasing function on integers such that $F(n) = F(n-1) + 2F(n-2)$ for all sufficiently large n. Find a number K such that $F(n) = \Theta(K^n)$.

Assume $F(n) = K^n$. Then $K^n = K^{n-1} + 2K^{n-2}$ for large n. Divide both sides by K^{n-2}
$K^2 = K + 2$
$K^2 - K - 2 = 0$
$(K-2)(K+1) = 0$
$K = 2, -1$
K cannot be negative because F is increasing, thus $K = 2$.

11. [10 points]

Write C++ code for selection sort. To get you started, I wrote the first line. Assume `swap(int a, int b)` is defined and exchanges its parameters.

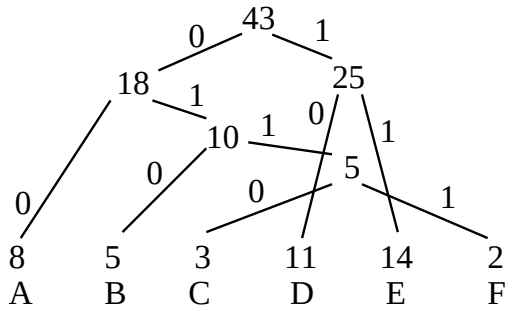
```
void sort(int x[n])
 {
  for(int i = 0; i < n-1; i++)
   for(int j = i+1; j < n; j++)
    if(x[i] > x[j])
     swap(x[i],x[j])
 }
```

12. [10 points] The following is a partial array implementation of the ADT stack of integer. Fill in the missing code for empty and push.

```
struct stak
 {
  int item[N];
  int size;
 };

void init(stak&S)
 {
  S.size = 0;
 }

bool empty(stak S)
 {
  return S.size == 0;
 }

void push(stak&S, int newitem)
 {
  S.item[S.size] = newitem;
  S.size++;
 }
```

13. [10 points] Find an optimal binary prefix-free code on the alphabet A,B,C,D,E,F where frequencies are given in the following table.

A	8	00
B	5	010
C	3	0110
D	11	10
E	14	11
F	2	0111

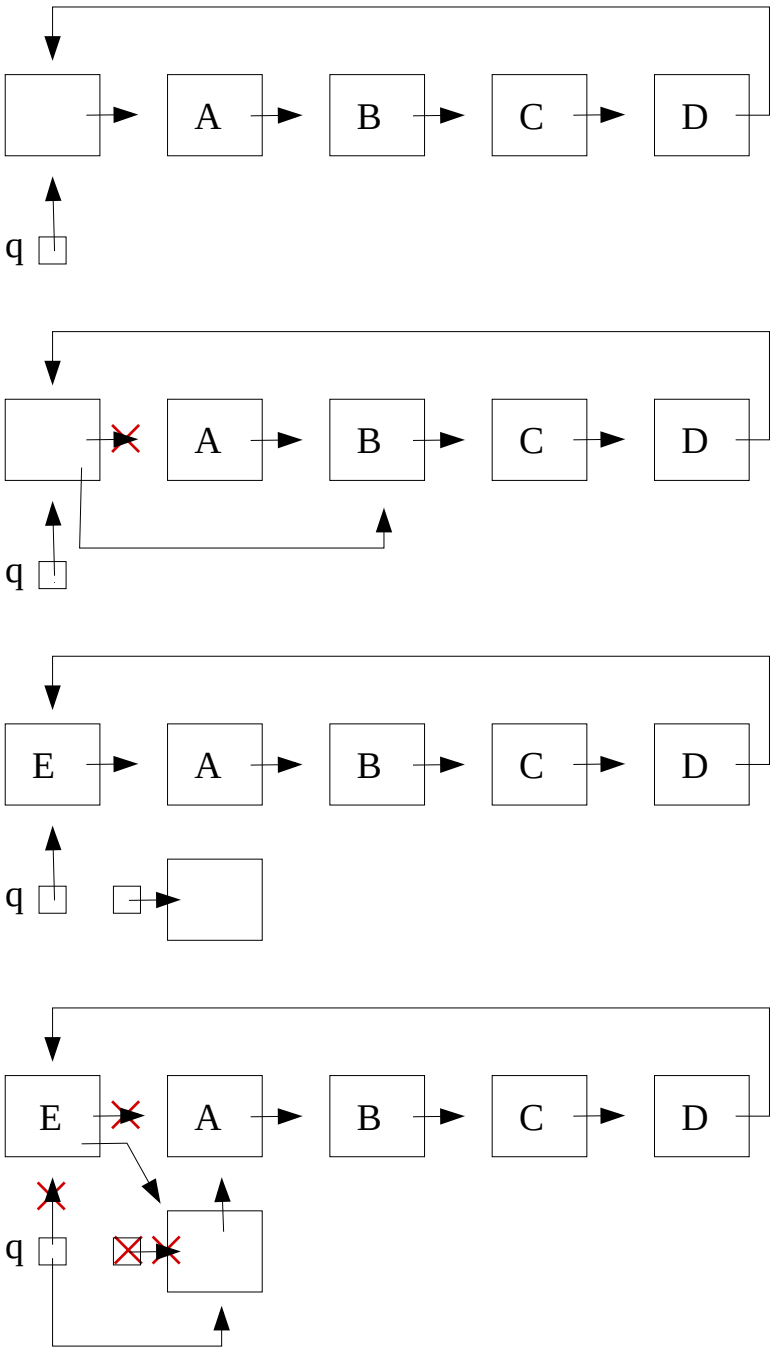


14. [10 points] Show the steps of polyphase mergesort, sorting the array AXFGUNTRELKSCVW
AXY,FGU,NT,R,EL,KS,CVW AXY,NT,EL,CVW
FGU,R,KS
AFGUXY,EKLS
NRT,CVW
AFGNRTUXY
CEKLSVW
ACEFGKLNRSTUVWXY

15. [10 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 A, B, C, D.

(a) Show the steps of dequeue.

(b) Redraw the original figure, and show the steps of enqueue(E).



In each case, do not erase deleted objects. Instead, cross them out.