

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

Final Examination May 8, 2023

Name: \_\_\_\_\_

No books, notes, scratch paper, or calculators. Use pen or pencil, any color. Use the rest of this page and the backs of the pages for scratch paper. If you need more scratch paper, it will be provided. If you want anything on extra pages to be graded, staple those pages to your test and write, "Please grade this page."

The entire examination is 400 points.

1. In each blank, write  $\Theta$  if correct, otherwise write  $O$  or  $\Omega$ , whichever is correct. [5 points each]

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

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

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

(iv)  $\log_2 n = \text{_____} (\log_4 n)$

(v)  $n^{0.000000000001} = \text{_____} (\log n)$

(vi)  $\log^* \log n = \text{_____} (\log^* n)$

2. True or False. Write "T" or "F." If the answer is not known to science at this time, write "O" for "Open." [5 points each]

(i) \_\_\_\_\_ The subproblems of a dynamic program form a directed acyclic graph.

(ii) \_\_\_\_\_ Kruskal's algorithm uses dynamic programming.

(iii) \_\_\_\_\_ A hash function should appear to be random, but cannot actually be random.

3. Solve each recurrence, expressing the answer as an asymptotic function of  $n$ . Use  $O$ ,  $\Omega$ , or  $\Theta$ , whichever is most appropriate. Although I haven't in previous tests, this time I will take a point off for expressing the answer without using asymptotic notation, or for using the wrong one. [5 points each]

(i)  $F(n) \geq 3F(n/9) + 1$

(ii)  $F(n) = F(3n/5) + 4F(2n/5) + n^2$

(iii)  $F(n) = 2F(n/2) + n$

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

(v)  $F(n) \geq F(n/2) + 2F(n/4) + n$

(vi)  $F(n) \leq 4F(n/2) + n^2$

(vii)  $F(n) \leq F(\sqrt{n}) + 1$

(viii)  $F(n) = 2F(n/4) + \sqrt{n}$

4. Give the asymptotic complexity, in terms of  $n$ , for each of these code fragments. [5 points each]

- (i) 

```
for(int i = 0; i < n; i++)  
    for(int j = n; j > i; j = j/2)
```
- (ii) 

```
for(int i = 0; i < n; i++)  
    for(int j = i; j > 0; j = j/2)
```
- (iii) 

```
for(int i = 2; i < n; i=i*i)
```
- (iv) 

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

5. Fill in the blanks.

- (i) [5 points] \_\_\_\_\_ is a fast implementation of selection sort.
- (ii) [5 points] \_\_\_\_\_ is a fast implementation of insertion sort.
- (iii) [5 points] The recurrence  $F(n) = F(n/5) + F(7n/10) + n$  is used to compute the time complexity of \_\_\_\_\_.
- (iv) [5 points] A \_\_\_\_\_ hash function fills the hash table exactly with no collisions.
- (v) [5 points] \_\_\_\_\_ algorithm finds a binary code for a weighted alphabet such that the code for one symbol is never a prefix of the code for another symbol.
- (vi) [5 points] The asymptotic expected height of a treap with  $n$  nodes is \_\_\_\_\_.
- (vii) [5 points] If  $G$  is a weighted digraph, it is impossible to solve any shortest path problem on  $G$  if  $G$  has a \_\_\_\_\_.
- (viii) [5 points] The height of a binary tree with 45 nodes is at least \_\_\_\_\_. (You must give the exact answer. No partial credit.)
- (ix) [5 points] The following is pseudo-code for what algorithm?

```
_____  
  
int x[n];  
input values of x;  
for(int i = n-1; i > 0; i--)  
    for(int j = 0; j < i; j++)  
        if(x[i] < x[j]) swap(x[i],x[j]);
```

- (x) [5 points] \_\_\_\_\_ algorithm does not allow the weight of any arc to be negative.
- (xi) [5 points] The asymptotic time complexity of Johnson's algorithm on a weighted directed graph of  $n$  vertices and  $m$  arcs is \_\_\_\_\_. (Your answer should use  $O$  notation.)
- (xii) [5 points] The time complexity of every comparison-based sorting algorithm is \_\_\_\_\_. (Your answer should use  $\Omega$  notation.)

- (xiii) [10 points] The prefix expression  $*a+ \sim b * -c d \sim e$  is equivalent to the infix expression \_\_\_\_\_ and the postfix expression \_\_\_\_\_.
- (xiv) [5 points] A \_\_\_\_\_ hash function fills the hash table exactly with no collisions.
- (xv) [5 points] In \_\_\_\_\_ there can be any number of items at a given index of the hash table.
- (xvi) [5 points] In closed hashing, if the position at  $h(x)$  is already occupied for some data item  $x$ , a \_\_\_\_\_ sequence is used to find an unoccupied position in the hash table.
- (xvii) [5 points] In \_\_\_\_\_ hashing, each item has more than one hash value, but only uses one of them.
- (xviii) [5 points] A planar graph with  $n \geq 3$  vertices can have no more than \_\_\_\_\_ edges. (Exact formula, please.)
6. [10 points] What is the loop invariant of the loop in the following function?

```
float product(float x, int n)
{
    // assert(n >= 0);
    float z = 0.0;
    float y = x;
    int m = n;
    while(m > 0)
    {
        if(m%2) z = z+y;
        m = m/2;
        y = y+y;
    }
    return z;
}
```

7. [10 points] The following code is used as a subroutine for both quicksort and select. Assume  $A[n]$  is an array of integers. For simplicity, we assume that no two entries of  $A$  are equal. Write a loop invariant for the while loop.

```
int pivot = A[0];
int lo = 0;
int hi = n-1;
while(lo < hi)
{
    if(A[lo+1] < pivot) lo++;
    else if(A[hi] > pivot) hi--;
    else swap(A[lo+1],A[hi]);
}
```

8. [10 points] Use dynamic programming to compute the Levenshtein edit distance between the strings “abcdef” and “bedafc.”

9. [10 points] Sketch a circular linked list with dummy node which implements a queue. The queue has four items. From front to rear, these are A, B, C, D, and show the insertion of E into the queue. Show the steps. Don't erase deleted objects; instead, simply cross them out.

10. [10 points] List properties of a good hash function used for a search structure.
11. [20 points] Write pseudo-code for the Floyd/Warshall algorithm. Let the vertices be  $\{1, 2, \dots, n\}$ . Let  $W(i, j)$  be the weight of the arc  $(i, j)$ , if any. Let  $W(i, j) = \infty$  if there is no arc. Compute  $V(i, j)$ , the minimum weight of any path from  $i$  to  $j$ , and  $B(i, j)$ , the backpointer for that minimum path.
12. [10 points] Walk through polyphase mergesort with the array given below.
- ACBXFREYGMQSNDZ**
13. [20 points] A compiler stores an array  $A[8][10][18]$  into main memory in row major order, with base address  $B$ , and each entry of  $A$  requires one place in main memory. Write a formula for the main memory address of  $A[i][j][k]$  for integers  $i, j$ , and  $k$  within range.

14. Consider the following C++ code.

```
int george(int n)
{
    if(n == 0) return 1;
    else return george(n/2)+george(n/2-1)+n*n;
}
```

- (a) [10 points] What is the asymptotic complexity of `george(n)`?
  - (b) [10 points] What is the asymptotic time complexity of the recursive code given above?
  - (c) [10 points] What is the asymptotic time complexity of a bottom-up dynamic programming algorithm to compute `george(n)`?
  - (d) [10 points] What is the asymptotic space complexity of a top-down computation of `george(n)` using memoization?
15. [20 points] Consider an array implementation of a stack of integers, as given below. Fill in the code which implements the needed operators of a stack.

```
const int N = // whatever
struct stack
{
    int item[N];
    int size; // number of items in the stack
    // bottom of the stack is at item[0];
};
void initialize(s&stack)
{

}
void push(s&stack,int i)
{

}
bool empty(s&stack)
{

}
int pop(s&stack)
{

}
```

16. [10 points] In class, we implemented a minheap as an almost complete binary tree implemented as an array. Suppose the minheap is initialized as shown in the first line of the array shown below. Show the evolution of the structure when deletemin is executed.

A	C	F	D	Q	H	L	R	Z

17. [10 points] Starting from the configuration given, show the evolution of the structure when B is inserted.

C	D	F	R	Q	H	L	Z	

18. [20 points] You are given an acyclic directed graph  $G = (V, E)$  where  $V = \{1, 2, \dots, n\}$ , and where each arc is weighted. If  $(i, j)$  is an arc, we write  $W(i, j)$  for the weight of that arc. Give a dynamic programming algorithm which calculates the directed path through  $G$  of maximum weight. For example, in the digraph shown below, the maximum weight directed path is  $(1, 2, 5, 6)$ . Assume that  $i < j$  for any arc  $(i, j)$ , as in the figure. You need not write pseudo-code if you can explain the algorithm without it.

