CSE 2320
Test 1
Spring 2019

Key 1
Your name as it appears on your UTA ID Card

Multiple Choice:

1. Write the letter or value of your answer on the line (_____) to the LEFT of each problem.
2. CIRCLED ANSWERS DO NOT COUNT.
3. 2 points each

1. The time to run the code below is in:

```
for (i=n; i>=0; i--)
  for (j=0; j<n*n; j+=2)
    sum+=i*j;
```

C A. $\Theta(n \log n)$ B. $\Theta(n^2)$ C. $\Theta(n^3)$ D. $\Theta(n)$

2. When did we use $\sum_{k=0}^{t} x^k \le \sum_{k=0}^{\infty} x^k = \lim_{k \to \infty} \frac{x^k - 1}{x - 1} = \frac{1}{1-x}$?

C A. To define H_n
B. For a recursion tree that has the same contribution for each level
C. For a recursion tree that has decreasing contributions by each level going away from the root
D. For a recursion tree that has increasing contributions by each level going away from the root

3. Which of the following is true?

A A. $n^3 \in \Omega(n^2)$ B. $n \log n \in \Omega(n^2)$
C. $g(n) \in O(f(n)) \Leftrightarrow f(n) \in O(g(n))$ D. $3^n \in O(2^n)$

4. Bottom-up maxheap construction is based on applying `maxHeapify` in the following fashion:

B A. In ascending slot number order, for each slot that is a parent.
B. In descending slot number order, for each slot that is a parent.
C. $\frac{n}{2}$ times, each time from subscript 1.
D. In descending slot number order, for each slot that is a leaf.

5. Which function is in both $\Omega(2^n)$ and $O(3^n)$, but is not in $\Theta(2^n)$ and $\Theta(3^n)$?

C A. $2^n + n^2$ B. $3^n - n^2$ C. 2.5^n D. $\ln n$

6. $f(n) = \lg n$ is in all of the following sets, except

C A. $O(\log n)$ B. $O(\log(n!))$ C. $\Omega(n)$ D. $o(n^2)$

7. Suppose the input to heapsort is always a table of n ones. The worst-case time will be:

A A. $\Theta(n)$ B. $\Theta(n^2)$ C. $\Theta(n \log n)$ D. $\Theta(\log n)$

8. The time to run the code below is in:

```
sum=1;
for (i=1; i<n*n; i=3*i)
  sum++;
```

C A. $\Theta(n \log n)$ B. $\Theta(n^2)$ C. $\Theta(\log n)$ D. $\Theta(n)$ E. $\Theta(3^n)$

9. The number of calls to `heapExtractMin` to build a Huffman code tree for n symbols is:

D A. $\Theta(\log n)$ B. $n - 1$ C. n D. $2n - 2$

10. Suppose you are using the substitution method to establish a Θ bound on a recurrence $T(n)$ and you already know $T(n) \in \Omega(n)$ and $T(n) \in O(n^2)$. Which of the following cannot be shown as an improvement?

A A. $T(n) \in O(\lg n)$ B. $T(n) \in O(n)$ C. $T(n) \in \Omega(n^2)$ D. $T(n) \in \Omega(n \lg n)$

11. Suppose a binary search is to be performed on a table with 62 elements. The maximum number of elements that could be examined (probes) is:

6

12. Heapsort may be viewed as being a faster version of which sort?

A A. selection B. `qsort`
C. insertion D. mergesort

13. The expected time for insertion sort for n keys is in which set? (All $n!$ input permutations are equally likely.)

D A. $\Theta(\log n)$ B. $\Theta(n)$ C. $\Theta(n \log n)$ D. $\Theta(n^2)$

14. Which of the following is a longest common subsequence for 0 1 2 0 1 2 and 0 0 1 1 2 2?

B   A. 0 0 2 2   B. 0 1 2 2   C. 0 0 1 1   D. 0 0 1 1 2

15. The time to extract the LCS (for sequences of lengths m and n) after filling in the dynamic programming matrix is in:

B   A. $\Theta(n)$ B. $\Theta(m+n)$ C. $\Theta(n \log n)$ D. $\Theta(mn)$

16. What is the value of $\sum_{k=0}^{\infty} \left(\frac{1}{3}\right)^k$?

$\frac{3}{2}$

17. The time to multiply two $n \times n$ matrices is:

C   A. $\Theta(n)$ B. $\Theta(\max(m,n,p))$ C. $\Theta(n^3)$ D. $\Theta(mnp)$

18. The goal of the optimal matrix multiplication problem is to:

C
A. Minimize the number of C(i,j) instances evaluated.
B. Minimize the number of matrix multiplications.
C. Minimize the number of scalar multiplications.
D. Minimize the number of scalar additions.

19. Which of the following is solved heuristically by a greedy method?

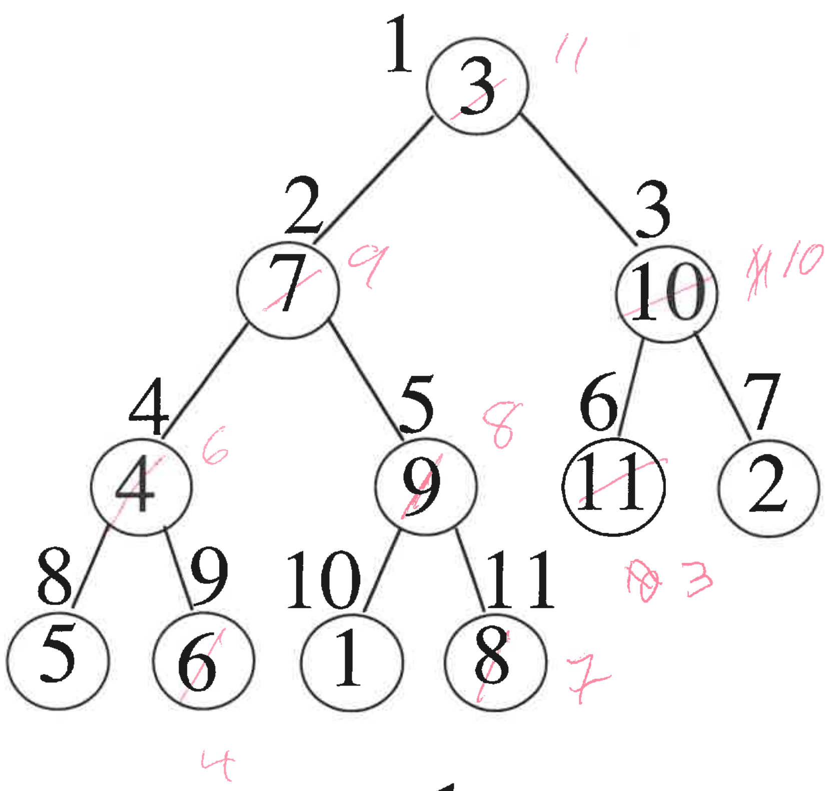
D
A. Fractional knapsack   B. Huffman coding
C. Unweighted interval scheduling   D. 0/1 knapsack

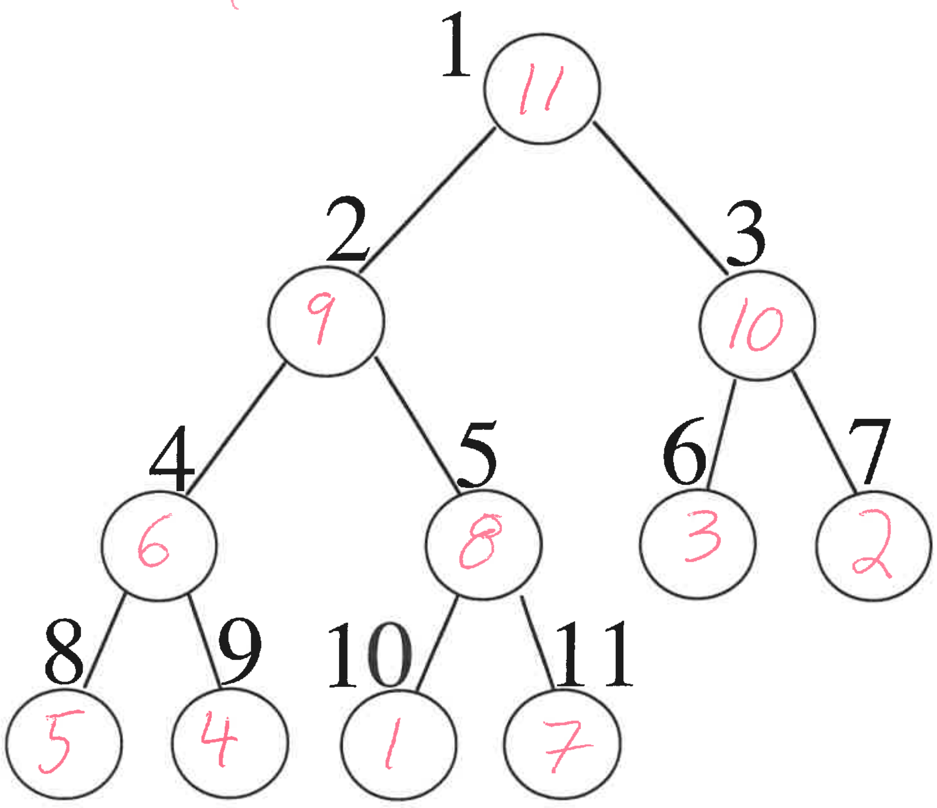
20. Suppose a Huffman code tree is constructed for an alphabet with eight symbols where each symbol has a probability of 0.125 of occuring. What is the expected bits per symbol?

3

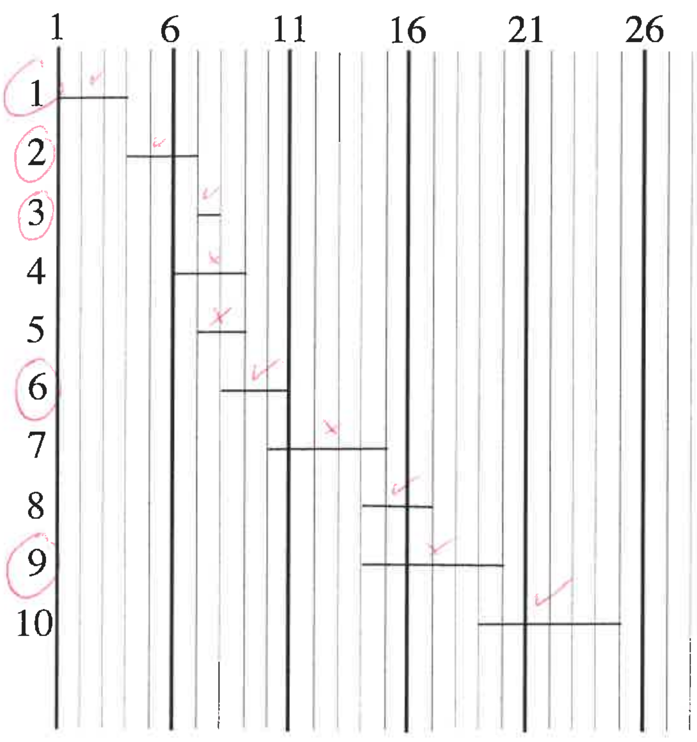
Long Answer

1. Use the efficient construction from Notes 05 to convert into a maxheap. 10 points





2. Use dynamic programming to solve the following instance of weighted interval scheduling. Be sure to clearly indicate the intervals in your solution and the sum achieved. 10 points



	v_i	p_i	$m(i)$
1	1	0	1
2	5	1	6
3	3	2	9
4	8	1	9
5	1	2	9
6	2	3	11
7	3	5	12
8	1	6	12
9	4	6	15
10	1	8	15

3. Use the greedy method for unweighted interval scheduling for the set of intervals in the previous problem. You may give your solution as the numbers of the chosen intervals. 5 points

1, 2, 3, 6, 8, 10

4. set containment

```
i=j=0;
while (i<m && j<n)
  if (A[i]<B[j])
    return 0;
  else if (A[i]>B[j])
    j++;
  else
  {
    i++;
    j++;
  }
return i==m;
```

```
i=j=0;
while (i<m && j<n)
  if (A[i]==B[j])
    i++;
  else
    j++;
return i==m;
```

```
i=0;
for (j=0; i<m && j<n; j++)
  if (A[i]==B[j])
    i++;
return i==m;
```

```
j=0;
for (i=0;i<m;i++)
{
  for ( ;j<n && A[i]>B[j];j++)
    ;
  if (j==n || A[i]<B[j])
    return 0;
}
return 1;
```

```
int sc(int m,int n,int *A,int *B);
{
if (m==0)
  return 1;
if (n==0)
  return 0);
if (A[0]<B[0]
  return 0;
if (A[0]>B[0])
  return sc(m,n-1,A,B+1);
return sc(m-1,n-1,A+1,B+1);
}
```

```
i=j=C=0;
while (i<m && j<n)
  if (A[i]>B[j])
    j++;
  else if (A[i++]==B[j++])
    C++;
  else
    return 0;
return C==m;
```

```
int sc(int m,int n,int *A,int *B);
{
if (m==0)
  return 1;
if (n==0)
  return 0);
if (A[0]==B[0]
  return sc(m-1,n-1,A+1,B+1);
return sc(m,n-1,A,B+1);
}
```

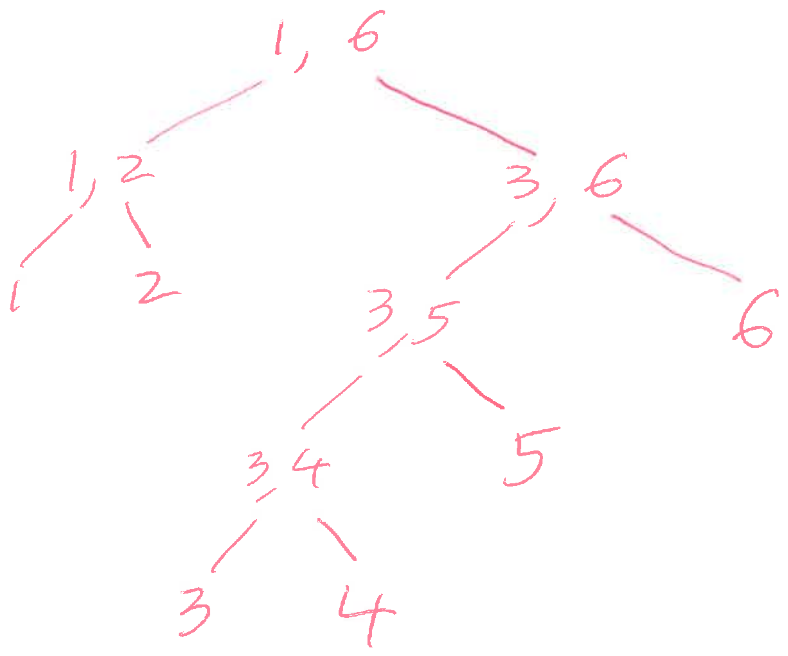
4. Two `int` arrays, `A` and `B` contain `m` and `n` `ints` each, respectively, with `m<=n`. The elements within both of these arrays appear in **ascending order** without duplicates (i.e. each table represents a set).

   Give `C` code for a $\Theta(m+n)$ algorithm to test **set containment** (`A` $\subseteq$ `B`) by checking that every value in `A` appears as a value in `B`. If set containment holds, your code should `return 1`. If an element of `A` does not appear in `B`, your code should `return 0`.

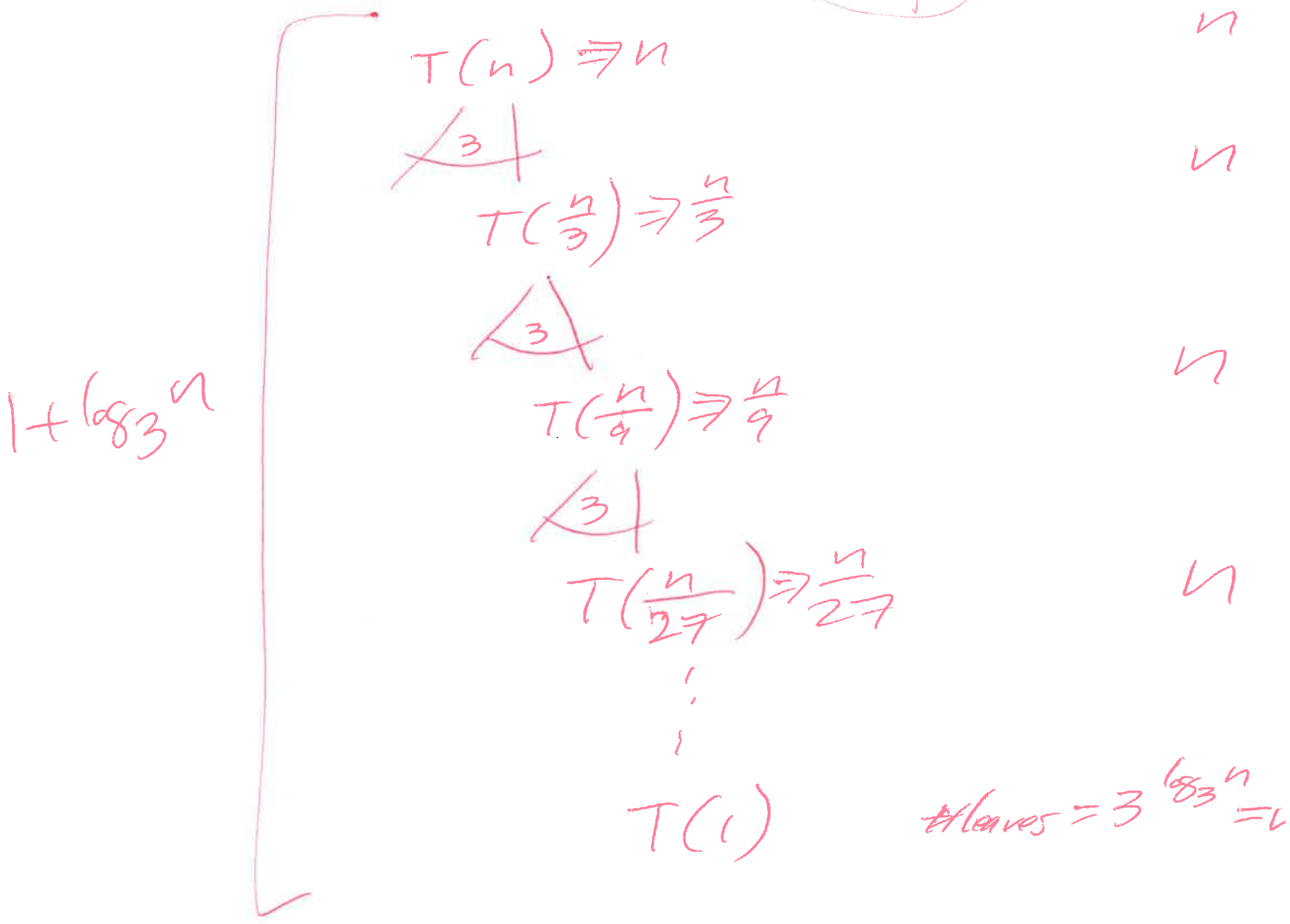
   (Details of input/output, allocation, declarations, error checking, comments and style **are unnecessary**.) 10 points

5. Give the tree corresponding to the following instance of optimal matrix multiplication. 5 points

```
6
4  3  2  5  4  7  5

             1          2          3          4          5          6
 1      0     0    24    1    64    2    96    2   176    2   230    2

 2   -------        0    0    30    2    64    2   138    2   196    2

 3   -------   -------         0    0    40    3    96    4   166    5

 4   -------   -------   -------          0    0   140    4   240    4

 5   -------   -------   -------   -------          0    0   140    5

 6   -------   -------   -------   -------   -------          0    0
```



6. Use the recursion-tree method to show that $T(n) = 3T\left(\frac{n}{3}\right) + n$ is in $\Theta(n \log n)$. 10 points



$T(n) \Rightarrow n$
3
$T(\frac{n}{3}) \Rightarrow \frac{n}{3}$
3
$T(\frac{n}{9}) \Rightarrow \frac{n}{9}$
3
$T(\frac{n}{27}) \Rightarrow \frac{n}{27}$
$T(1)$
$1 + \log_3 n$
n
n
n
n
~~#~~leaves $= 3^{\log_3 n} =$

$$(1 + \log_3 n)n = \Theta(n \log n)$$

7. Use the substitution method to show that $T(n) = 3T\left(\frac{n}{3}\right) + n$ is in $O(n \log n)$. (You do not need to show that $T(n)$ is in $\Omega(n \log n)$.) 10 points

Suppose $T(k) \le ck \log_3 k$ for $k < n$

$T\left(\frac{n}{3}\right) \le c\frac{n}{3} \log_3 \frac{n}{3}$

$= c\frac{n}{3}\left[\log_3 n - 1\right]$

$T(n) = 3T\left(\frac{n}{3}\right) + n$

$\le 3c\frac{n}{3}\left[\log_3 n - 1\right] + n$

$= cn \log_3 n \boxed{- cn + n}$

$\le cn \log_3 n$ for $c \ge 1$