CSE 2320
Test 1
Fall 2017

Key
Your name as it appears on your UTA ID Card

Multiple Choice:

1. Write the letter 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-1; i>=0; i-=2)
  for (j=5; j<n; j*=3)
    sum+=i+j;
```

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

2. A sort is said to be stable when:

A A. Items with the same key will appear in the same order in the output as in the input.
B. It runs in $O(n\log n)$ time.
C. The expected time and the worst-case time are the same.
D. It removes duplicate copies of any key in the final output.

3. Which of the following is false?

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

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

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

5. The function $2\log n + n$ is in which set?

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

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

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

7. Which of the following facts can be proven using one of the limit theorems?

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

C. $g(n) \in \Theta(f(n)) \Leftrightarrow f(n) \in \Theta(g(n))$ D. $3^n \in \Omega(2^n)$

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

```
for (i=n-5; i>=5; i--)
  for (j=2; j<n; j+=2)
    sum+=i+j;
```

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

9. Which sort takes worst-case $\Theta(n^2)$ time and is not stable?

D A. heap B. insertion C. merge D. selection

10. Suppose you are using the substitution method to establish a Θ bound on a recurrence $T(n)$ and you already know that $T(n) \in \Omega(1)$ and $T(n) \in O(n^2)$. Which of the following cannot be shown as an improvement?

D A. $T(n) \in O(1)$ B. $T(n) \in O(\log n)$ C. $T(n) \in \Omega(n^2)$ D. $T(n) \in \Omega(n^3)$

11. What is n, the number of elements, for the largest table that can be processed by binary search using no more than 7 probes?

D A. 31 B. 63 C. 64 D. 127

12. What is the definition of H_n?

D A. $\Theta(\sqrt{n})$ B. $\sum_{k=1}^{n} k$ C. $\ln n$ D. $\sum_{k=1}^{n} \frac{1}{k}$

13. Which of the following functions is not in $\Omega(n^2)$?

C A. $n^2 \lg n$ B. n^3 C. n D. n^2

14. $4^{\lg 5}$ evaluates to which of the following? (Recall that $\lg x = \log_2 x$.)

C A. $\sqrt{7}$ B. 7 C. 25 D. 49

15. When solving the fractional knapsack problem, the items are processed in the following order.

D
____ A. Ascending order of weight
B. Ascending order of $$$/lb
C. Descending order of weight
D. Descending order of $$$/lb

16. What is the value of $\sum_{k=0}^{\infty} \left(\frac{2}{3}\right)^k$?

D
____ A. $\frac{1}{3}$ B. $\frac{2}{3}$ C. $\frac{3}{2}$ D. 3

17. Suppose there is a large table with n integers, possibly with repeated values, in ascending order. How much time is needed to determine the number of occurences of a particular value?

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

18. The recursion tree for mergesort has which property?

C
____ A. it leads to a definite geometric sum
B. it leads to a harmonic sum
C. each level has the same contribution
D. it leads to an indefinite geometric sum

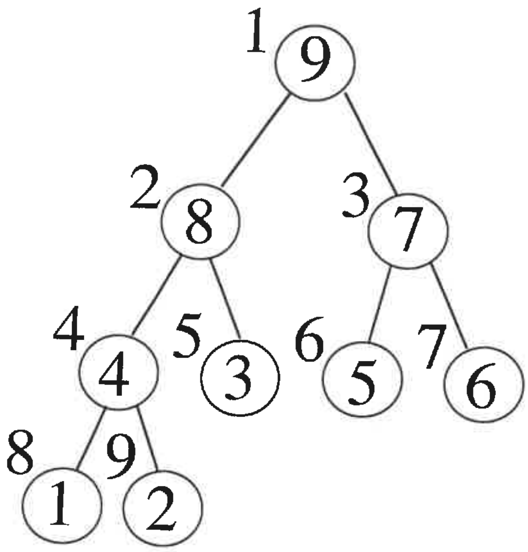
19. Which of the following is not true regarding a maxheap with 1000 elements?

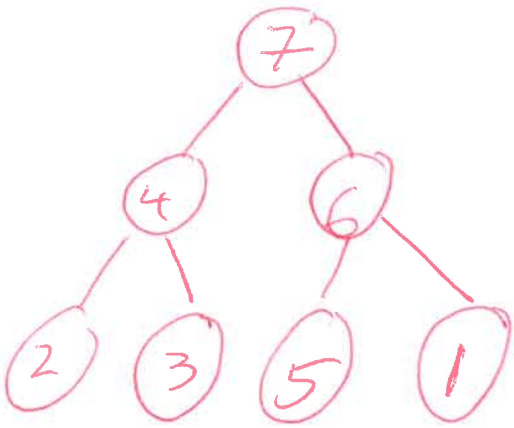
D
____ A. Subscript 1 will store the maximum priority.
B. The parent for the node with subscript 500 is stored at subscript 250.
C. The left child for the node with subscript 200 is stored at subscript 400.
D. The right child for the node with subscript 405 is stored at subscript 911.

20. The goal of the Huffman coding method is:

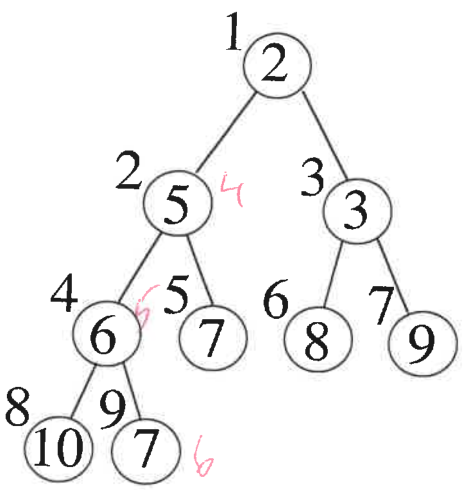
C
____ A. Construct a max-heap for the symbols in an alphabet
B. Find the symbols with high probability of occuring.
C. Minimize the expected bits per symbol.
D. Maximize the compression for every string.

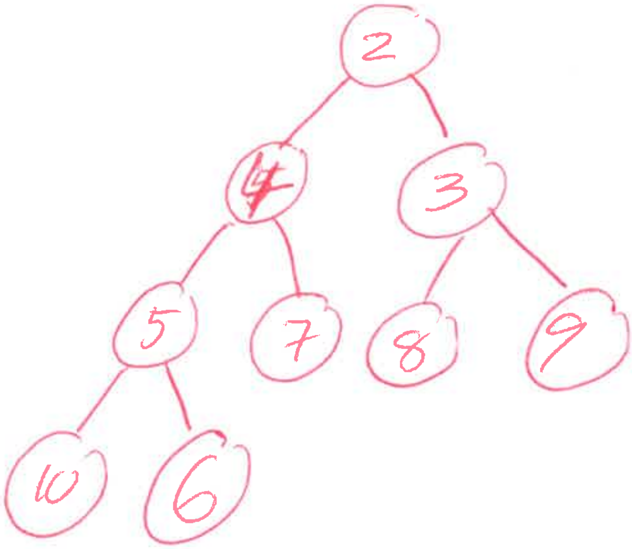
1. Show the result after performing `heapExtractMax` <u>twice</u> on the following maxheap. 5 points



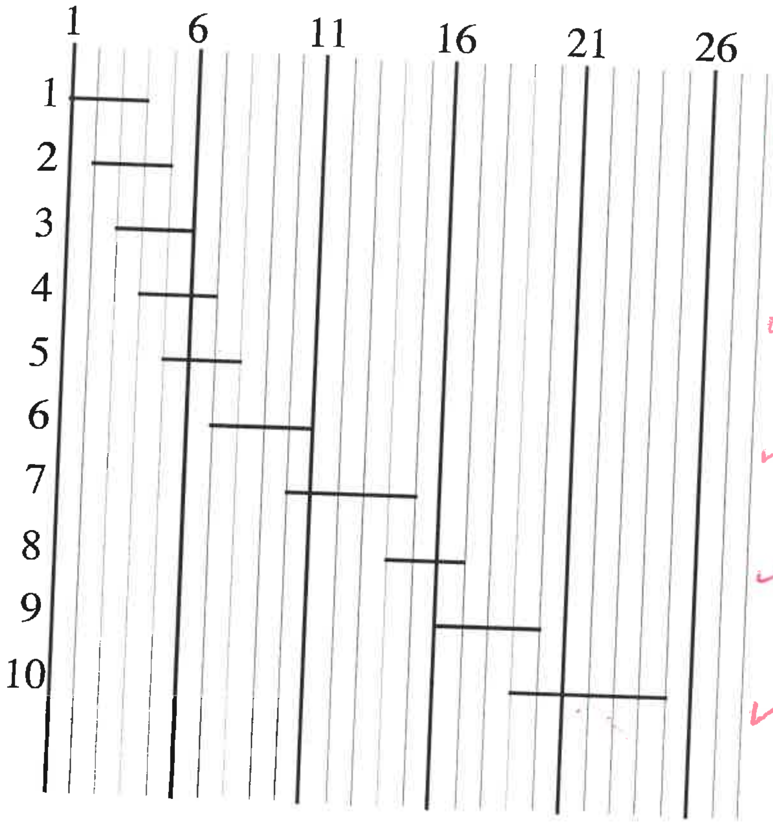


2. Show the minheap after changing (`minHeapChange`) the priority at subscript 9 to 4. 5 points





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



v	p	m
5	0	5
4	0	5
3	0	5
2	1	7
1	2	7
5	4	12
4	5	12
3	6	15
2	7	15
1	8	16

4 1. 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;
```

```
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;
```

```
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;
```

```
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;
```

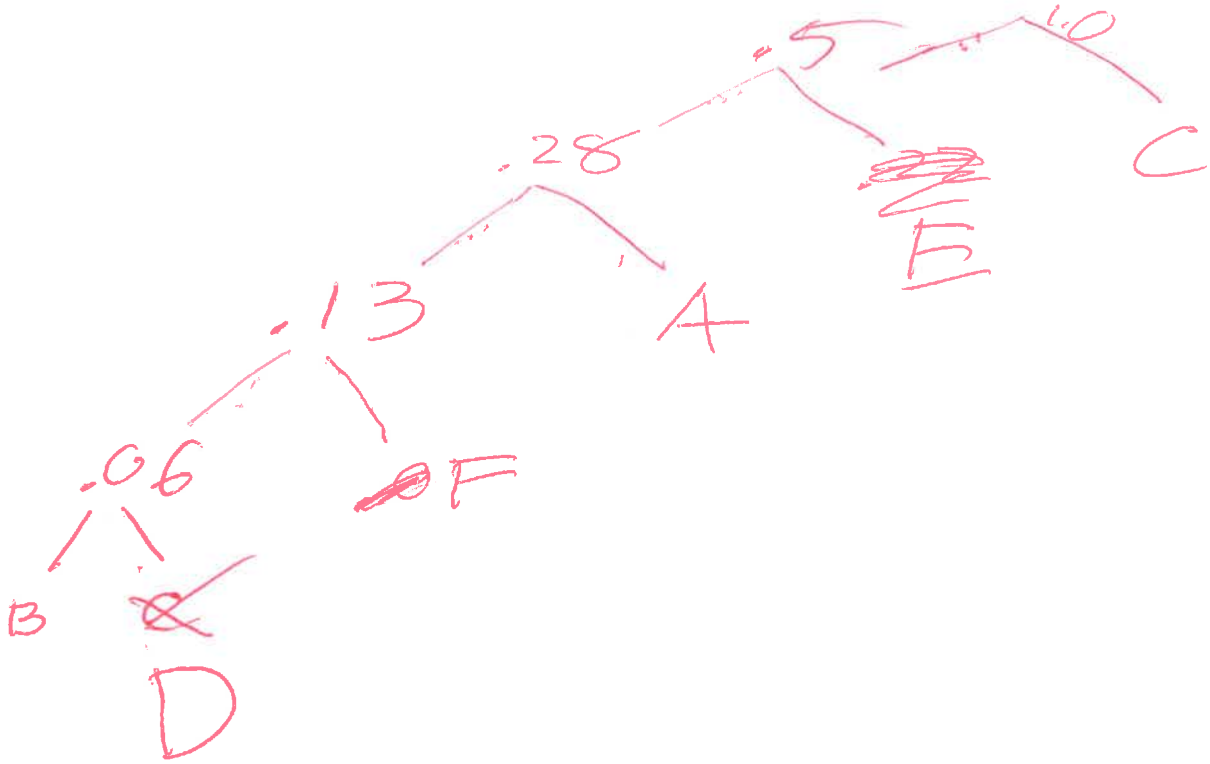
```
int sc(int m,int n,int *A,int *B);
{
if (m==0)
  return 1;
if (n==0)
  return 0);
if (A[0]<B[0]
  return 0;
if (A[0]>B[0])
  return sc(m,n-1,A,B+1);
return sc(m-1,n-1,A+1,B+1);
}
```

```
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



A	.15	3	.45
B	.03	5	.15
C	.5	1	.50
D	.03	5	.15
E	.22	2	.44
F	.07	4	.28
			1.97

5. Give a Huffman code tree for the following symbols and probabilities. Besides the tree, be sure to compute the expected bits per symbol. 10 points

A	0.15
B	0.03
C	0.5
D	0.03
E	0.22
F	0.07

$\Omega(n^2)$ not required

6. Use the recursion-tree method to show that $T(n) = 3T\left(\frac{n}{3}\right) + n^2$ is in $\Theta\left(n^2\right)$. 10 points

$T(n) \Rightarrow n^2$ $\qquad n^2$

3

$T\left(\frac{n}{3}\right) \Rightarrow \frac{n^2}{9}$ $\qquad \cdot 3 = \frac{n^2}{3}$

3

$T\left(\frac{n}{9}\right) \Rightarrow \frac{n^2}{81}$ $\qquad \cdot 9 = \frac{n^2}{9}$

3

$T\left(\frac{n}{27}\right) \Rightarrow \frac{n^2}{729}$ $\qquad \cdot 27 = \frac{n^2}{27}$

$1 + \log_3 n$

$T(1)$ $\qquad$ # leaves $= 3^{\log_3 n} = n$

$$n + n^2 \sum_{k=0}^{\log_3 n - 1} \left(\frac{1}{3}\right)^k \le n + n^2 \sum_{k=0}^{\infty} \left(\frac{1}{3}\right)^k$$

$$= n + n^2 \frac{1}{1 - \frac{1}{3}}$$

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

7. Use the substitution method to show that $T(n) = 3T\left(\frac{n}{3}\right) + n^2$ is in $O\left(n^2\right)$. (You do not need to show that $T(n)$ is in $\Omega\left(n^2\right)$.) 10 points

Suppose $T(k) \leq ck^2$ for $k < n$

$$T\left(\frac{n}{3}\right) \leq c\frac{n^2}{9}$$

$$T(n) = 3T\left(\frac{n}{3}\right) + n^2$$

$$\leq 3\frac{cn^2}{9} + n^2$$

$$= c\frac{n^2}{3} + n^2$$

$$= cn^2 - \frac{2}{3}cn^2 + n^2$$

$$\leq cn^2 \text{ for } c \geq \frac{3}{2}$$