Practice Final Exam

Due No due date **Points** 60 **Questions** 49 **Time Limit** None **Allowed Attempts** 3

Instructions

For practice only. Not graded. You get three attempts.

Take the Quiz Again

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	467 minutes	60 out of 60

! Correct answers are hidden.

Score for this attempt: **60** out of 60

Submitted Jun 7 at 8:27am

This attempt took 467 minutes.

Question 1

1 / 1 pts

Rank the following functions by increasing order of growth:

$\log(n^2)$, $10000n$, n^3, 2^n, $\sqrt[3]{n}$

- $\log(n^2)$, n^3, 2^n, $\sqrt[3]{n}$, $10000n$
- $\sqrt[3]{n}$, $\log(n^2)$, $10000n$, n^3, 2^n
- $10000n$, $\log(n^2)$, n^3, 2^n, $\sqrt[3]{n}$
- $10000n$, $\log(n^2)$, n^3, $\sqrt[3]{n}$, 2^n,
- none of the above

Question 2

1 / 1 pts

Give a tight bound on the number of times the statement z = z + 1 is executed. *Assume n is a power of 2.*

```
i = n

while ( i > 1 ) {

  i = floor(i/2)

  z = z + 1

}
```

- Theta(n)
- Theta(nlgn)

Theta(sqrt(n))

Theta(lgn)

Theta(1)

Question 3

1 / 1 pts

Determine the theoretical running time of the following algorithm:

```
int Algo1(int n)
{
    int sum = 0;
    for (int i = n; i > 0; i--) {
        for (int j = i+1; j <=n; j++) {
            sum = sum + j;
            cout << i << " " << sum << endl;
        }
    }
}
```

$\Theta(n)$

$\Theta(\lg n)$

$\Theta(n^2)$

$\Theta(nlgn)$

None of the above

Question 4

2 / 2 pts

Let $f(n) = n^3$

Let $g(n) = n^2 \log(n^3)$

What is the asymptotic relation between f(n) and g(n)? **Check all that apply.**

- [ ] f(n) = O(g(n))
- [x] f(n) = Omega(g(n))
- [ ] f(n) = Theta(g(n))
- [ ] g(n) = Theta(f(n))
- [x] g(n) = O(f(n))

g(n) = Omega(f(n))

Question 5

3 / 3 pts

Let $f(n) = 10000n^3$

Let $g(n) = n^4$

What is the asymptotic relation between f(n) and g(n)? **Check all that apply.**

- [x] f(n) = O(g(n))
- [ ] f(n) = Omega(g(n))
- [ ] f(n) = Theta(g(n))
- [ ] g(n) = Theta(f(n))
- [ ] g(n) = O(f(n))
- [x] g(n) = Omega(f(n))

Question 6

1 / 1 pts

If $g(n) = 2^n$ and $f(n) = 2^{n+1}$. What is the asymptotic relation between f(n) and g(n)? **Check all that apply.**

- [x] $f(n) = O(g(n))$
- [x] $f(n) = \Omega(g(n))$
- [x] $f(n) = \Theta(g(n))$
- [x] $g(n) = \Theta(f(n))$
- [x] $g(n) = O(f(n))$
- [x] $g(n) = \Omega(f(n))$

Question 7

1 / 1 pts

If $f(n) = O(n^2)$ and $g(n) = O(n^2)$, then $f(n) = \Theta(g(n))$.

- ( ) True
- (•) False

consider, e.g., f(n) = n^2 and g(n) = n

Question 8

1 / 1 pts

Given the recursive algorithm Foo(n) below, write a recurrence, T(n), for its running time.

```
Foo(n) {
  total = 0
  if n = 1 return 2
  else {
      total = Foo(n/4) + Foo(n/4)
      for i = 1 to n do
          for k = 1 to 3 do
              total = total + k
       return total  }
}
```

- ( ) $T(n) = T(n/2) + 2n$
- (•) $T(n) = 2T(n/4) + 3n$
- ( ) $T(n) = 2T(n/4) + n^2$
- ( ) $T(n) = T(n/4) + 2n^2$

None of the above

Question 9

1 / 1 pts

Given the following algorithm

```
goo(n)
  if n <= 1 {
     return 1 }
 else {
    x = goo(n-1)
    for i = 1 to 3*n {
        x = x + i }
    return x }
```

Determine the asymptotic running time. Assume that addition can be done in constant time.

- $\Theta\left(n^2\right)$
- $\Theta\left(n\right)$
- $O(n)$

- $\Theta(2^n)$
- $\Theta(n^3)$
- none of the above

Question 10

1 / 1 pts

Solve the following recurrence by giving the tightest bound possible.

$$T(n) = 2T(\frac{n}{8}) + 4n^2$$

- $\Theta(n^2)$
- $\Theta(n)$
- $\Theta(n\log(n))$
- $\Theta(\log(n))$
- None of the above

Question 11

1 / 1 pts

Solve the following recurrence by giving the tightest bound possible.

$$T(n) = 4T(\frac{n}{4}) + 4n$$

- $\Theta(logn)$
- $\Theta(n^2)$
- $\Theta(nlgn)$
- $\Theta(n)$
- $\Theta(n^3)$
- None of the above

Question 12

1 / 1 pts

Solve the following recurrence by giving the tightest bound possible.

T(n) =T(n-1) + 5

- $\Theta(logn)$

- $\Theta(n)$
- $\Theta(n^3)$
- $\Theta(n^2)$
- None of the above

Question 13

1 / 1 pts

Solve the following recurrence by giving the tightest bound possible.

$$T(n) = T(n-2) + n$$

- $\Theta(n^2)$
- $\Theta(n)$
- $\Theta(n\log(n))$
- $\Theta(\log(n))$
- None of the above

Question 14

1 / 1 pts

Which of the following is/are property/properties of a dynamic programming problem?

- Optimal Substructure
- Overlapping Subproblems
- Greedy approach
- Both optimal substructure and overlapping subproblems

Question 15

1 / 1 pts

Consider the following two sequences :

X = < K, L, M, L, J, K, L >, and

Y = < L, J, M, K, L, K >

The length of the longest common subsequence of X and Y is :

- 5
- 3

- 4
- 2
- 1

Question 16

1 / 1 pts

If a dynamic programming algorithm uses an nxn table then the running time is **always :**

- O(n^2)
- O(n)
- O(nlgn)
- O(n^3)
- not enough information to determine

Question 17

1 / 1 pts

All dynamic programming problems can be solved by using a greedy choice algorithm.

- True
- False

Question 18

5 / 5 pts

Suppose we have an alphabet with only five letters A, B, C, D , E which occur with the following frequencies:

- A = 51, B = 10, C = 8, D = 12, E = 19

Construct a Huffman code using the following guidelines while constructing the code

- the lowest frequency node is the left child in the tree while the higher frequency node is the right child
- when creating the code the left branch is assigned a 0 while the right branch is assigned a 1.

The Huffman binary coding is:

- A = 11, B = 110, C = 010, D = 1100, E = 0011
- A = 1, B = 0111, C = 0110, D = 101, E = 11
- A = 1, B = 0111, C = 0110, D = 010, E = 00

A = 0, B = 11, C = 10, D = 11, E = 111

A = 0, B = 1,1, C = 1001, D = 101, E = 1000

Question 19

1 / 1 pts

Consider the following greedy choice strategies to solve the activity-selection problem of section 16.1 in CLRS.

Select the compatible activity with :

1. the earliest start time.
2. the shortest total time.
3. the fewest conflicts.
4. the latest finishing time.
5. the latest start time.

Which strategy is guaranteed to result in an optimal solution.

1

2

3

4

- 5

Question 20

1 / 1 pts

Consider weights and values of items below and a knapsack that can hold at most 20 lbs.

Item	Value in $	Weight in lbs
1	15	10
2	30	15
3	48	12
4	25	5
5	12	4

Assume that each item can be used at most once and **can be broke**n. What is the maximum value of items that can be placed in the knapsack.

- 90
- 85
- 82
- 87

- None of the above

Question 21

1 / 1 pts

Let X be an NP-complete problem and Y and Z be two other problems not known to be in NP. Y is polynomial time reducible to X and X is polynomial-time reducible to Z. Which of the following statements is true?

- Z is in NP-complete
- Z is in NP-Hard
- Y is in NP-complete
- Y is in NP-Hard

Question 22

1 / 1 pts

The problems 3-SAT and 4-SAT are

- Both in P

Both in NP-Hard but not in NP.

Both in NP-complete

None of the above

Question 23

1 / 1 pts

Consider two decision problems X and Y. If X reduces in polynomial time to 3-SAT and 3-SAT reduces in polynomial time to Y. Which of the following can be inferred from the previous statement?

X is in NP and Y is in NP-Hard

Y is in NP and X is in NP-Hard

Both X and Y are in NP-hard.

Both X and Y are in NP.

Question 24

1 / 1 pts

A problem in NP is in NP-complete if

- It can be reduced to CIRCUIT-SAT in polynomial time
- It can be reduced to all problems in NP-complete.
- The 3-SAT problem can be reduced to it in polynomial time.
- Some problem in P can be reduced to it.

Question 25

1 / 1 pts

If you discover a polynomial time algorithm for the SUBSET-SUM problem this will imply that P=NP.

- True
- False
- This is unknown

Question 26

1 / 1 pts

NP-complete is a subset of NP-Hard.

True

False

Question 27

1 / 1 pts

If you discover a polynomial time algorithm for the 0-1 knapsack problem this will imply that P=NP.

True

False

Question 28

1 / 1 pts

Every problem in P can be reduced to HAM-CYCLE,

True

Every problem in P is in NP, and every problem in NP can be reduced to any NP-complete problem. Circuit-SAT is NP-complete.

False

Question 29

1 / 1 pts

The traveling salesman problem can never be solved exactly.

- True
- False

Question 30

1 / 1 pts

Approximation algorithms are used to solve NP-complete decision problems.

- True
- False

Question 31

1 / 1 pts

Given two vertices s and t in a connected graph G, which of the two traversals, BFS and DFS can be used to find if there is a path from s to t?

- Only DFS
- Only BFS
- Both BFS and DFS
- Neither BFS nor DFS

Question 32

1 / 1 pts

Let T be a complete binary tree with n vertices . Finding a shortest path (measured by number of edges) from the root of T to a given vertex $v \in T$ takes

- O(n)
- O(lgn)
- O(n^2)
- O(nlgn)

Question 33

1 / 1 pts

Let G be a graph with n vertices and m edges. Assume that the graph is represented by an adjacency matrix. What is the tightest upper bound on the running time of DFS performed on G?

- ◉ O(n^2)
- ○ O(m+n)
- ○ O(mn)
- ○ O(m)
- ○ O(n)

Question 34

3 / 3 pts

Given a weighted directed graph G = (V,E,w) and a shortest path P from s to t, if we doubled the weight of every edge to produce G'=(V,E,w'), then P is also a shortest path in G'.

- ◉ True
- ○ False

Question 35

1 / 1 pts

Dijkstra’s algorithm may not terminate with the correct distances if the graph contains negative-weight cycles.

- (selected) True
- False

Question 36

1 / 1 pts

In an undirected graph with edge weights that are all 1, a DFS from vertex s to some vertex t will always produce a shortest path from s to t.

- True
- (selected) False

Question 37

1 / 1 pts

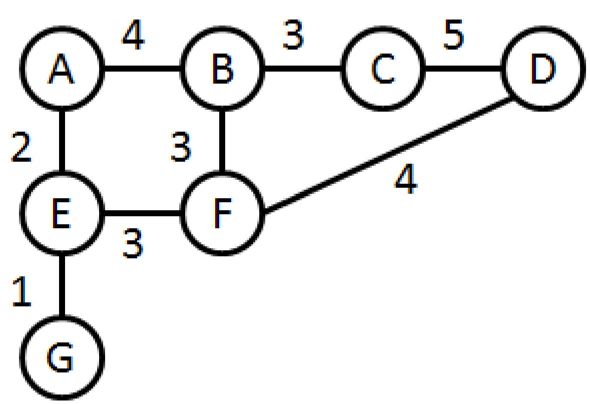
In an undirected weighted graph the heaviest edge is never in the MST.

- True
- False

Question 38

1 / 1 pts

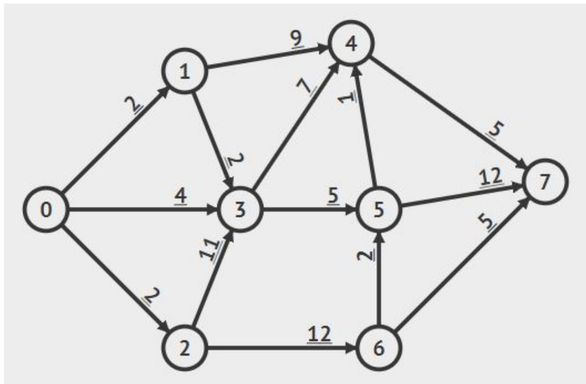
What is the weight of the MST for the graph below? Give strictly a numeric answer.



16

Question 39

1 / 1 pts



In the graph above, the shortest path from vertex 0 to vertex 7 has weight of

- 21
- 19
- 15
- 16

Question 40

3 / 3 pts

Given a weighted directed graph G = (V,E,w) and a shortest path P from s to t, if we doubled the weight of every edge to produce G'=(V,E,w'), then P is also a shortest path in G'.

True

False

Question 41

1 / 1 pts

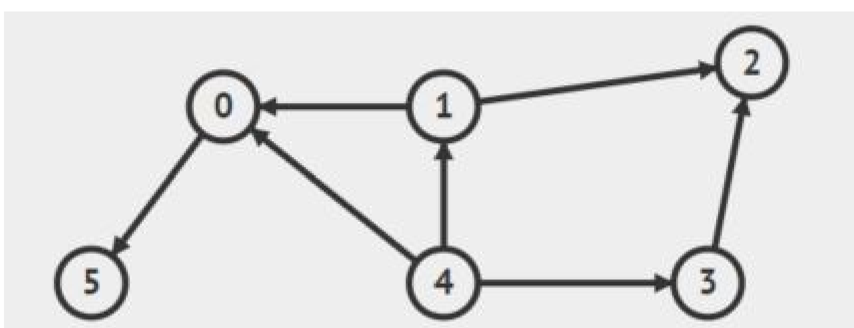
Dijkstra’s algorithm may not terminate with the correct distances if the graph contains negative-weight cycles.

True

False

Question 42

1 / 1 pts

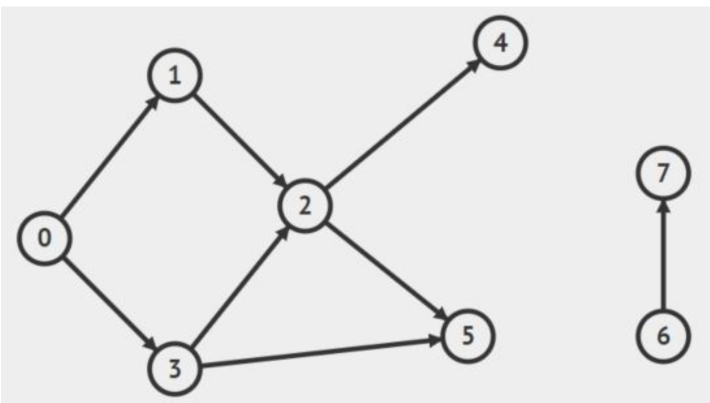


A Breadth First Search Algorithm has been implemented using a queue data structure. One possible order of visiting the vertices of the graph above is:

- 0, 5, 4, 1, 2
- 4, 1, 0, 5, 2, 3
- 4, 0, 1, 3, 5, 2
- 1, 0, 5, 2, 4, 3

Question 43

1 / 1 pts



Which of the following is a topological sort of the graph above.

- 7, 6, 5, 2, 4, 3, 1, 0

- 6, 7, 0, 1, 3, 2, 5, 4
- 0, 1, 2, 3, 4, 5, 6, 7
- 0, 3, 2, 5, 1, 4, 6, 7
- None of the above

Question 44

1 / 1 pts

What is the running time of a 2-OPT approximation algorithm for the 0-1 Knapsack problem that uses the greedy criteria of value/weight?

- Theta(n)
- Theta(nlgn)
- Theta(lgn)
- Theta(n^2)
- None of the above

Question 45

1 / 1 pts

Which of the following graph algorithms can be used to create a polynomial-time 2-approximation algorithm for the traveling salesman problem?

- ○ DFS
- ○ BFS
- ○ Shortest Path
- ◉ MST
- ○ None of the above

Question 46

1 / 1 pts

An an approximation algorithm with an approximation ratio of 2 is always twice as fast as an exact algorithm for solving the problem.

- ○ True
- ◉ False

Question 47

1 / 1 pts

In the TSP problem with Euclidean distances what is the relationship between the cost of the MST and the cost of the optimal TSP tour T*? *Assume all edge wrights are positive.*

- cost(MST) = cost(T*)
- cost(MST) > cost(T*)
- cost(MST) < cost(T*)
- None of the above

Question 48

1 / 1 pts

You are using a polynomial time 2-approximation algorithm to find a tour t for the traveling salesman problem. Which of the following statements is true.

- The tour t is never optimal.
- The cost of tour t is at most twice the cost of the optimal tour.e the
- The cost of tour t is always 2 times the cost of the optimal tour.

The ratio of the cost of the optimal tour divided by the cost of tour t is 2.

All of the above

Question 49

1 / 1 pts

```
Testing(n) {
  total = 0
  if n = 1 return 2
  else {
      total = Testing(n/4) + Testing(n/4)
      for i = 1 to n do
            for k = 1 to n do
                  total = total + k
       return total  }
}
```

Write a recurrence for the running time T(n) of Testing(n)

- $T(n) = T(n/4) + n$

- $T(n) = T(2n/4) + n^2$
- $T(n) = 2T(n/4) + n^2$
- $T(n) = 2T(n/4) + n$
- None of the above

Quiz Score: **60** out of 60