Do 6 of the following 8 questions (omit 2). You have two hours to finish.

1. For each of the following recurrence relations obtain a simple closed-form formula for the order of \( t(n) \), i.e., \( \Theta(t(n)) \). Be sure to show your steps.
   a) \( t(n) = 3t(\lfloor n/3 \rfloor) + n \), \hspace{1cm} \text{init. cond.} \ t(1) = 1.
   b) \( t(n) = t(n) + 1/n \), \hspace{1cm} \text{init. cond.} \ t(0) = 50.

2. a) Write a high-level linear-time algorithm for computing the \( \sqrt{n} \)-smallest element of an \( n \) element list efficiently. You may use randomization.
   b) Explain how to use parallelism to speed up your algorithm from part (a).

3. a) Describe a parallel algorithm for sorting a list \( L[1:n] \) of \( n \) elements on the two-dimensional mesh \( M_{q,q} \), where \( n = q \times q \).
   b) Analyze the performance (complexity, speed, cost) of the algorithm you have given in part (a).

4. Consider the change-minimization problem. There are coin denominations of \( (25, 10, 5, 1) \). For a given value \( x \), we need to determine the minimum number of coins whose value totals \( x \).
   a) Describe the greedy algorithm for this problem.
   b) Give a set of coin denominations and a value for \( x \) such that the greedy algorithm fails to produce the minimum change required.
   c) Suppose that coin denominations were \( (100, 10, 1) \). Does the greedy algorithm give an optimal solution? Prove your answer is correct.

5. Suppose \( F \) is a part of a finding a minimum weight spanning tree, and \( \alpha \) is the smallest weighted edge that has one end in \( F \) and the other end is not in \( F \).
   a) Prove that \( F \cup \{\alpha\} \) is also part of a minimum weight spanning tree.
   b) Use the result of part (a) to design an algorithm to find a minimum weight spanning tree.

6. A unimodal sequence is a list of \( n \) numbers that first strictly increases and then strictly decreases. Present an algorithm to find the maximum in a unimodal sequence which has worst-case complexity \( W(n) = \Theta(\log n) \).

7. a) Describe how the Fast Fourier Transform (FFT) can be employed to multiply polynomials efficiently.
   b) Given that \( DFT_4(P(x)) \) and \( DFT_4(Q(x)) \) have coefficient arrays \( (2,1+i,0,1-i) \) and \( (3,2+i,1,2-i) \), respectively, compute \( P(x)Q(x) \) using FFT.

8. a) Give pseudocode for Floyd’s all-pairs shortest path algorithm.
   b) Briefly explain how Floyd’s algorithm can be used to compute longest paths in dags (directed acyclic graphs).
You have two hours. Do 6 of the following 8 questions. Omit 2.

1. a) Does \(1 + 2 + \ldots + n = \Theta(n)\)? Discuss.
   
b) Show that \(P(n) \in \Theta(n^k)\), for \(P(n)\) a polynomial of degree \(k\).
   
c) Show that \(O(a^n) \subset O(b^n)\) for \(1 < a < b\).

2. a) What is \(\sum_{i=0}^{\infty} (1/2)^i\)?
   
b) What is \(\sum_{i=0}^{\infty} i(1/2)^i\)?
   
c) What is the depth (height) of the complete binary tree having \(2^k\) nodes?
   
d) What is the harmonic series? Give a closed form approximation of it.
   
e) Give an order formula for \(\log_2 1 + \log_2 2 + \ldots + \log_2 n\). Prove your formula.

3. For each of the following recurrence relations obtain a closed-form formula for \(t(n)\). Show steps.
   
a) \(t(n) = t(n-1) + n\), init. cond. \(t(0) = 0\).
   
b) \(t(n) = 7t(n/2)\), init. cond. \(t(1) = 1\) (where \(n\) is a power of 2).

4. a) Prove that the depth \(D(T)\) of a binary tree on \(N\) nodes is at least \(\log_2(N+1) - 1\).
   
b) Using the result in part (a) show that \(\Omega(n \log n)\) is a lower bound for the problem of comparison-based sorting.

5. Explain how the Fast Fourier Transform (FFT) algorithm can be used to multiply two polynomials of size \(n\) in time \(O(n \log n)\).

6. a) Describe a parallel algorithm to add \(n\) numbers on the 2-dimensional mesh \(M_{qq}\), \(n = q^2\).
   
b) Analyze the performance (complexity, speedup, cost) of your algorithm from part (a).
   
c) Prove that any algorithm for summing \(n\) numbers on \(M_{qq}\) must have communication complexity at least \(2\sqrt{n} - 2\).

7. A graph is 2-colorable if each node of the graph can be marked with one of two colors (read and blue), such that each pair of adjacent nodes have different colors.
   
a) Using induction prove that all trees are 2-colorable.
   
b) Give a linear-time algorithm which will test whether an arbitrary graph (implemented using adjacency lists) is 2-colorable.

8. a) Give the definition of a (binary max-) heap
   
b) The nodes of a heap are stored in an array level by level, beginning at the root and stored left to right within each level. Show by induction that the left child of the \(i\)th node is indexed by \(2i\).
   
c) Draw the following array as a complete binary tree and identify the node or nodes that are not the root of a heap.

   \[
   20,10,7,6,5,3,8,7,2,3,2,1.\]
Design & Analysis of Algorithms Part of Qualifying Exam    Fall 1995

Do 6 of the following 10 questions. Omit 4. You must explain your answers to receive full marks.

1. Consider the algorithm Linear Search whose input is a list \( L[1..n] \) of size \( n \) and a search element \( X \), and whose output is the index of the first occurrence of \( X \) or 0 if \( X \) is not in the list.
   a) What are the best and worst case complexities of Linear Search?
   b) Assume that \( L \) is a list of \( n \) distinct elements, and that the probability that \( X \) is found in the list is \( 2/3 \). Also assume that, given that \( X \) is in the list, there is an equally likely chance of \( X \) occurring in any position. Determine the average case complexity of Linear Search.

2. a) Using the definition of \( \Theta \) show that \( (2n^3 - 4n^2 + 10)2^n \in \Theta(n^32^n) \).
   b) Using limits and L’Hopital’s rule show that \( O(n^4) \subset O(2^n) \).
   c) Give an order formula for each of the following, i.e. give the representative of the appropriate \( \Theta \)-class whose form is the simplest:
      
      (i) \( 1^7 + 2^7 + 3^7 + \ldots + n^7 \)
      
      (ii) \( \frac{1}{5} + \frac{1}{6} + \ldots + \frac{1}{n} \)
      
      (iii) \( 1 + 2n + 3n^2 + \ldots + (k+1)n^k \)
      
      (iv) \( \left[ \frac{n}{7} \right] \)
      
   d) Does \( \log_2(2) + \log_2(3) + \ldots + \log_2(n) = \Theta(\log_2 n) \)? Discuss.

3. a) Give pseudocode for MergeSort. Do not include pseudocode for procedure Merge.
   b) Assuming that the worst case complexity of procedure Merge is \( n - 1 \), obtain a recurrence relation for the worst case complexity \( W(n) \) of Mergesort. You may assume the \( n = 2^k, k \geq 0 \).
   c) Solve the recurrence relation you have given in b) to obtain an explicit formula for the worst case complexity of Mergesort.

4. a) Give pseudocode for the sorting algorithm Quicksort.
   b) Describe the tree of recursive calls for an input of size \( n \) to Quicksort yielding the best case complexity \( B(n) \), and give an order formula for \( B(n) \).
   c) Give the tree of recursive calls for an input of size \( n \) to Quicksort yielding the worst case complexity \( W(n) \), and give an order formula for \( W(n) \).
5. a) Assume the nodes of a binary tree are represented by dynamic variables having the structure:

\[ \text{TreeNode} = \text{record} \]
\[ \quad \text{info: integer} \]
\[ \quad \text{LeftChild: } \rightarrow \text{TreeNode} \]
\[ \quad \text{RightChild: } \rightarrow \text{TreeNode} \]
\[ \text{end TreeNode} \]

Give pseudo-code for a recursive function `Depth(Root)` that computes the depth of the binary tree having root pointed to by `Root`.

b) Give pseudocode for the algorithm `DFT(G)` that performs the depth-first traversal of graph `G`. Include pseudocode for the recursive algorithm `DFS(G,v)` called by `DFT`.

6. Let `G` be any connected graph and let `T` be any spanning tree of `G`.

   a) Let `e` be any edge of `G` not belonging to `T`. Prove that the subgraph `T ∪ \{e\}` contains a unique circuit `C`.

   b) Let `e'` be any edge of the circuit `C` from part a). Prove that the subgraph `(T ∪ \{e\}) \setminus \{e'\}` is a spanning tree.

   c) Find a minimum cost spanning tree in the weighted graph below.

\[ G = \]

![Graph Diagram]

7. Consider the recurrence relation:

\[ t(n) = 6t(n-1) - 9t(n-2), \forall n \geq 2, \ t(0) = 0, \ t(1) = 1. \]

   a) Obtain a closed-form formula for the associated generating function

\[ g(x) = t(0) + t(1)x + \ldots + t(n)x^n + \ldots \]

   b) Solve the above recurrence relation for `t(n)` by finding the power series expansion of the formula for `g(x)` you derived in a).
8. a) Give an $O(n^{\log_2 3})$ Divide-&-Conquer algorithm $PolyMult(P(x), Q(x), R(x))$ that computes the product polynomial $R(x)$ of two polynomials $P(x)$ and $Q(x)$ of size $n$.

b) Give a complexity analysis of the algorithm you have given in a).

c) Explain how a polynomial $P(x) = a_8 x^8 + \ldots + a_1 x + a_0$, and a quadratic $Q(x) = b_2 x^2 + b_1 x + b_0$ can be multiplied with the aid of repeated calls to $PolyMult$ (without padding $Q(x)$ with zeros).

9. The Discrete Fourier Transform of $P(x)$ with $\omega = i$, $DFT_i(P(x))$, has coefficient array $(0, -4+2i, -4, -4-2i)$. Find $P(x)$, using the Fast Fourier Transform algorithm.

10. a) Explain what is meant by the Principle of Optimality in Dynamic Programming.

   b) Show that the Principle of Optimality holds for binary search trees.

   c) Consider the names:

   Anne, Bob, Jack, Louis, Pete, Zak, Zelda.

   Assuming that there is an equally likely chance of searching for any of the seven names above, give an optimal binary search tree for these names.

   d) Using dynamic programming find an optimal binary search tree for the following probabilities:

   \[
   \begin{array}{c|cccc}
   keys & 1 & 2 & 3 & 4 \\
   \hline
   probabilities & p_i & .4 & .3 & .2 & .1
   \end{array}
   \]
Algorithms Part of Qualifying Exam

Please answer all seven questions. You have two hours to finish

1. a) State the definition of $\Omega(g(n))$.
   b) Using the definition, show that $n/2 - \log_2(n) = \Omega(n)$.
   c) Show that $3n \neq \Omega(n^2)$.

2. a) State the definition of a (max) heap.
   b) Given a heap, an adversary replaces the root entry with a number that is less than at least one child of the root. Write the pseudocode for reheapifying the heap. Use references such as leftchild(node) etc.
   c) Write the pseudocode for HeapSort using the reheapifying function of part b). Assume that the heap is already built at the start of the sorting process.

3. a) The fibonacci sequence is defined by the recurrence relation $f(n) = f(n-1) + f(n-2)$; $f(0) = 1$; $f(1) = 1$. Let $G(z) = \sum_{i=0}^{\infty} f(i)z^i$ be the generating function of the fibonacci sequence. State a recursive expression for $G(z)$.
   b) Find a closed form expression for $G(z)$.
   c) Find a partial fraction expansion of the expression of part b).
   d) Find a closed form expression for the $n^{th}$ fibonacci number from the result of part c).

4. a) Write a recurrence relation for the average complexity of QuickSort assuming all permutations are equally likely.
   b) Use induction and the result of part a) to show that the average time complexity of QuickSort is $O(n \log(n))$.

5. This problem pertains to the Greedy Method.
   a) Given a weighted undirected graph $G = (V, E)$ and weight function $w : E \rightarrow \mathbb{N}^+$, a spanning tree for $G$ is a subgraph $G' = (V, E')$ of $G$ such that $G'$ has no cycles and there is a path between every pair of vertices $v, w \in V$ using edges in $E' \subset E$. The weight of a spanning tree is the sum of the weights of its edges. Write pseudocode for a Greedy algorithm (Kruskal's) that finds a minimum weight spanning tree for a given graph $G$. 

1
b) Prove that the algorithm above is correct (you may use matroids, if you wish, but it is not necessary to do so).

6. a) Consider a binary search tree with the following labels on nodes: “m” on the root, “d” and “q” on the children of “m”, and “b” and “e” on the children of “d”. If a search key is chosen uniformly from the set of letters in the English alphabet, what is the average number of comparisons per search for the above search tree?

b) Consider a binary search tree of $2n + 1$ nodes ($n$ internal nodes and $n + 1$ leaves). Let $p_1, ..., p_n$ be the probabilities associated with internal nodes ($p_i$ is the probability that the key associated with the $i$th internal node is search key) and let $q_0, q_1, ..., q_n$ be the probabilities associated with the leaves. Let $d_i$ denote the depth of the $i$th internal node, $1 \leq i \leq n$. Let $e_i$ denote the depth of the $i$th leaf, $0 \leq i \leq n$. Write an expression for the average number of comparisons per search of this tree in terms of $d_i, e_i, p_i,$ and $q_i$.

c) Repeat part b) except write the expression in terms of $p_i, q_i,$ and the average search time for the left and right subtrees of the root.

d) Let $T_{ij}$ denote an optimal search tree (lowest average search time for the set of given keys and probabilities) involving the consecutive keys $i, ..., j$, where $T_{ij}$ is the null tree if $i > j$. Write an expression for $A(T_{ij})$, the average search time for $T_{ij}$, that can be used in a dynamic program to find the optimal search tree for all keys.

e) The dynamic program fills a table where the $ij$ element is $A(T_{ij})$. What element holds the result for all keys?
Design and Analysis of Algorithms
Qualifying Exam — November 11, 1994
Please answer all questions. You have two hours to finish.

1. Answer each of the following
   (a) What is \( \sum_{i=0}^{\infty} 2^{-i} \)?
   (b) What is \( \sum_{i=0}^{\infty} i2^{-i} \)?
   (c) How many nodes exactly are in a complete binary tree of height \( h \)?
   (d) How many nodes exactly are in a complete ternary tree of height \( h \)?
   (e) What is the harmonic series? Give an closed form approximation of it.
   (f) Show that if you square each term of the harmonic series the sum is bounded by a constant.
   (g) Find a closed form expression for \( A_n \) given by
       \[
       A_n = 4A_{n-2} \quad A_0 = 1, A_1 = 0
       \]

2. A graph is 2-colorable if each node of the graph can be marked with one of two colors (red and blue), such that each pair of adjacent nodes have different colors.
   a) Using induction prove that all trees are 2-colorable.
   b) Give a linear-time algorithm which will test whether an arbitrary graph (presented as an adjacency list) is 2-colorable.
3. Given a graph $G = (V, E)$, a vertex cover $C \subseteq V$ is defined as a set of nodes for which every edge in the graph has at least one of its two end-point nodes in $C$.

a. What is the smallest vertex cover of each of the following graphs?

b. How many sets of vertices would a naive algorithm have to look at to find the smallest vertex cover of an $n$-node graph?

c. Describe a greedy algorithm that should give a small, but not necessarily smallest vertex cover.

d. First, argue that the following algorithm always produces some vertex cover, and second, argue that the cover it returns is of size at most twice the optimal. Hint: find a property belonging to the set of edges that are picked in line 3.
1. $C := \emptyset$;
2. $E' := E$;
3. while ($E' \neq \emptyset$) do pick arbitrary edge $(u,v)$ in $E'$; add $u$ and $v$ to $C$; remove from $E'$ all edges incident to either $u$ or $v$; end while;
4. return $C$;
4. a) Give a lower bound on the number of comparisons to verify a list is sorted. Prove it.

b) Give a lower bound on the number of comparisons to sort an unordered list. Prove it.

c) Give a lower bound on the number of comparisons to find the max in an unordered list. Prove it.

d) Give a lower bound on the number of comparisons to find the max and the second max in an unordered list. Prove it.

e) Give an algorithm to find the max and the second max in an unordered list that matches the bound.
Design and Analysis of Algorithms
Qualifying Exam — April 1, 1994
Please answer all questions. You have two hours to finish.

1. a) State the definition of \( f(n) \in \Theta(g(n)) \).

   b) Show that \( \frac{1}{2}n^2 + n + 2 \in \Theta(n^2) \).

   c) Show that \( \sqrt{n} \notin \Theta(\log n) \).

2. Use "Big-Theta" notation \( \Theta \) for expressing each of the following values. Give the best answer possible.

   a) Number of comparisons in the worst case when sorting an \( n \) item list using Insertion Sort.

   b) Number of comparisons in the worst case when sorting an \( n \) item list using Quick Sort.

   c) Number of comparisons in the average case when sorting an \( n \) item list using Insertion Sort.

   d) Number of comparisons in the average case when sorting an \( n \) item list using Quick Sort.
e) The number of nodes in a complete binary tree of height \( h \)?

f) The number of nodes in a complete ternary tree of height \( h \)?

g) \( \sum_{i=1}^{n} \frac{1}{i} \)

h) \( \sum_{i=1}^{n} \frac{1}{i^2} \)

i) The solution to the recurrence \( T(n) = 2T(n/2) + 2n; \ T(1) = 0 \).

j) The solution to the recurrence \( T(n) = T(n/5) + T(3n/4) + 5n; \ T(1) = 0 \).

k) The solution to the recurrence \( T(n) = 3T(n - 2); T(0) = 1, T(1) = 1 \).

3. A hash function maps each \( n \)-bit key \( x \) to a table entry indexed by a \( p \)-bit number (where \( n > p \)). What is wrong with each of the following hash functions?

a. a random selection of \( p \) bits making up \( x \).

b. \( x \mod 2^p \).

c. \( x \div 2^{n-p} \).

Finally, define a reasonable hash function.

4. The nodes of a heap are stored in an array level by level, beginning at the root and left to right within each level. Show by induction that the left child of the \( i^{th} \) node is indexed by \( 2i \). (Hint: draw a picture.)
5. Draw the following array as a full binary tree and identify the node or nodes whose subtree is not a heap.

    20,10,7,6,5,3,8,7,2,3,2,1

6. Suppose that \( G = (V,E) \) is the undirected graph given by the following sets:

\[
V = \{a,b,c,d,e\}
\]

\[
E = \{\{a,b\},\{b,c\},\{e,d\},\{a,c\},\{c,d\}\}
\]

We input \( G \) to the following algorithm.

\[
T = \emptyset
\]

Repeat

Select the leftmost edge \( e = \{x,y\} \) from \( E \)

Remove \( e \) from \( E \)

If \( T \cup \{e\} \) is acyclic then \( T = T \cup \{e\} \)

until \( E = \emptyset \)

At every iteration of this algorithm, some nodes are connected by edges in \( T \) and some are not. Suppose we maintain disjoint subsets of nodes such that, for all iterations of the algorithm, any pair of nodes in \( V \) are in the same subset iff that pair are connected by at least one path of edges in \( T \). Such a structure allows a quick test for cycles as needed in the algorithm.

a. Give the structure for each iteration of the algorithm, one line per iteration.

b. Suppose we use an upward pointing set-of-trees structure to implement the structure defined above. In this set-of-trees structure UNIONs are done in the usual way (smaller tree attaches to bigger tree) and FINDs are done without path compression. Draw this set-of-trees structure for each iteration of the algorithm, one line per iteration.
7. The following is an essay question. Your answer should fit on one page. What is meant by a dynamic programming algorithm? How does it differ from greedy and divide-and-conquer paradigms? Define a problem and a solution algorithm that can be considered a use of dynamic programming.
Design and Analysis of Algorithms
Qualifying Exam — April 3, 1993
Please answer all questions. You have two hours to finish.

1. Describe an algorithm for finding a minimum weight spanning tree. Illustrate this algorithm for the sample graph below.

2. Solve the following recurrence relation $T(n) = 4T(n - 2); T(0) = 1, T(1) = 0.$
3. a) State the definition of $f(n) \in O(g(n))$.

b) Show that $10n^2 + n + 20 \in O(n^2)$.

c) Show that $3n^2 \not\in O(n)$.

4. Use “Big-Theta” notation $\Theta$ for expressing each of the following values

a) Number of comparisons in Worst case sequential search of an $n$ item list ...

b) Number of comparisons in Average case binary search of an $n$ item list ...

c) The sum of the first $n$ positive integers ...

d) The sum of the reciprocals of the first $n$ positive integers ...

e) The logarithm of the product of the first $n$ positive integers ...
5. a) Write a recurrence relation for the average complexity of QuickSort assuming all permutations are equally likely.

b) Solve the recurrence relation to show that the average time complexity of QuickSort is $\Theta(n \log(n))$. 
6. A unimodal sequence is a list of \( n \) numbers that first strictly increases and then strictly decreases. Present an algorithm to find the maximum in a unimodal sequence which has worst-case complexity \( W(n) = \Theta(\log n) \).
Design and Analysis of Algorithms
Qualifying Exam — November 11, 1993
Please answer all questions. You have two hours to finish.

1. Describe Prim's algorithm for finding a minimum weight spanning tree. Illustrate this algorithm for the sample graph below.

2. Solve the following recurrence relation. Be sure to prove your answer is correct.
   \[ T(n) = 2T\left(\lfloor n/2 \rfloor \right) + n; T(1) = 1 \]
3. a) State the definition of $f(n) \in \Omega(g(n))$.

b) Show that $\frac{1}{2}n^2 + n + 20 \in \Omega(n^2)$.

c) Show that $3n \not\in \Omega(n^2)$.

4. Use "Big-Omega" notation $\Omega$ for expressing each of the following values. Give the best answer possible.

a) Number of comparisons in worst case sequential search of an $n$ item list.

b) Number of comparisons in worst case search of a sorted $n$ item list.

c) The sum of the logarithms of the first $n$ positive integers.

d) The sum of the reciprocals of the first $n$ positive integers (i.e., $1 + 1/2 + 1/3 + ... + 1/n$).

e) The sum of the squares of the first $n$ positive integers.
5. a) Write a high-level linear-time algorithm for computing the median of a list efficiently. You may use randomization.

b) Explain how you can remove the randomization to achieve a deterministic linear-time algorithm.

6. Consider the change-minimization problem. There are coin denominations of (25, 10, 5, 1). For a given value \( x \), we need to determine the minimum number of coins whose value totals \( x \).

[a.] Describe the greedy algorithm for this problem.

[b.] Give a set of coin denominations and a value for \( x \) such that the greedy algorithm fails to produce the minimum change required.

[c.] Give a dynamic programming solution for the problem given by the coin denominations defined in b. Give an expression for the run time.
(a(b : 10)(c : 12)(d : 5)(e : -)),
(b(c : 2)(d : 16)(e : 10)),
(c(d : -)(e : 9)),
(d(e : 8)),

which means that a is connected to b by an edge of weight 10, and so on. A dash means there is no edge connecting the corresponding nodes. What is the minimum weight spanning tree of this graph?

b) State a Greedy method for finding it.

7. a) The Partition problem is stated as follows: Given a set of n objects \( S = \{ s_1, s_2, ..., s_n \} \), a function \( w : S \to \mathbb{N}^+ \) specifying the positive integer weights of individual objects, and a partition integer \( B \); find \( S' \subseteq S \) such that \( \sum_{s \in S'} w(s) = B \). Write a dynamic program for solving the Partition problem.

b) What is the complexity of your program?

8. This question pertains to NP-completeness.

a) Precisely state what is meant by the Bin Packing Problem.

b) What is the First-Fit approximation algorithm for Bin Packing and give a bound on its worst-case approximation performance.

c) Present an example which shows that First-Fit cannot do any better than \( \frac{A_{FF}(J)}{OPT(J)} = \frac{5}{3} \).
1. a) State the definition of \( f(n) = O(g(n)) \).
   
b) Using the definition, show that \( 3n + \log(n) = O(n) \).
   
c) Show that \( 3n^2 \neq O(n) \).

2. a) State the definition of a heap.
   
b) Given a heap, an adversary replaces the root entry with a number that is less than both descendent of the root. Write the pseudo-code for reheapifying the heap. Use references such as leftson(node) etc.
   
c) Write the pseudo-code for HeapSort using the reheapifying function of part b). Assume that the heap is already built at the start of the sorting process.

3. a) Write a recursive algorithm for InsertionSort.
   
b) Write a recurrence relation that expresses the worst case complexity of the algorithm of part a).
   
c) Solve the recurrence relation of part b).

4. a) The fibonacci sequence is defined by the recurrence relation \( f(n) = f(n-1) + f(n-2) \); \( f(0) = 1; f(1) = 1 \). Let \( G(z) = \sum_{i=0}^{\infty} f(i)z^i \) be the generating function of the fibonacci sequence. State a recursive expression for \( G(z) \).
   
b) Find a closed form expression for \( G(z) \).
   
c) Find a partial fraction expansion of the expression of part b).
   
d) Find a closed form expression for the \( n^{th} \) fibonacci number from the result of part c).

5. a) Write a recurrence relation for the average complexity of QuickSort assuming all permutations are equally likely.
   
b) Use induction and the result of part a) to show that the average time complexity of QuickSort is \( O(n \log(n)) \).

6. a) A weighted, undirected graph with nodes \( a, b, c, d, \) and \( e \) is specified by the following adjacency lists:
Do 6 of the following 8 questions. Omit 2.

1. a) Using the definition of \( \Theta \) show that \( 3n^4 - 7n^2 + 2 = \Theta(n^4) \).
   b) Let \( a, b \) be positive integers, \( a < b \). Show that \( O(a^n) \subset O(b^n) \), i.e.,
   \( a^n = O(b^n) \) but \( a^n \neq \Omega(b^n) \).
   c) Using limits and L'Hopital's rule show that \( O(\ln n) \subset O(n) \).
   d) Does \( 1^2 + 2^2 + \ldots + n^2 = \Theta(n^2) \)? Discuss.

2. a) Give the pseudo-code for Binary Search whose input is a list \( L[1..n] \) of size \( n \) and a search element \( X \), and whose output is the index of \( X \) or 0 if \( X \)
   is not in the list.
   b) What are the best and worst case complexities of Binary Search. Explain.

3. a) Give an efficient divide-and-conquer algorithm for multiplying two large
   integers \( u \) and \( v \) of size \( n \) (i.e. having \( n \) digits).
   b) Give a complexity analysis of the algorithm you have given in a).

4. a) Explain what is meant by the Principle of Optimality in Dynamic Programming.
   b) Show that the Principle of Optimality holds for shortest paths in graphs.
   c) Consider the names: Abe, Betty, Joe, Karl, Mary, Tom, Zeus.
   Assuming that there is an equally likely chance of searching for any of the
   seven names above, give an optimal binary search tree for these names.
   d) Using dynamic programming find an optimal binary search tree for the
   following probabilities:

   \[
   \begin{array}{c|cccc}
   keys & i & 1 & 2 & 3 & 4 \\
   \hline
   \text{probabilities} & p_i & .1 & .2 & .3 & .4 \\
   \end{array}
   \]

5. a) Give pseudo-code for quicksort. Do not include pseudo-code for the
   procedure pivot (also called partition).
   b) Describe the tree of recursive calls for an input of size \( n \) to quicksort
   yielding the best case complexity \( B(n) \), and give an order of magnitude
   formula for \( B(n) \).
   c) Give the tree of recursive calls for an input of size \( n \) to quicksort
   yielding the worst case complexity \( W(n) \), and give an order of magnitude
   formula for \( W(n) \).
6. a) Using Prim's algorithm find a minimum cost spanning tree $T$ in the weighted graph $G$ below. Mark the edges of $T$ directly on the graph $G$.

![Graph G]

b) Using Dijkstra's algorithm find a shortest path spanning tree $T$ rooted at node 1 in the weighted digraph $D$ below, i.e., $T$ contains a minimum length path from node 1 to all the other nodes where the length of a path is the sum of the weights of its edges. Mark the edges of $T$ directly on the digraph $D$. Also label each node with the length of a shortest path from 1 to that node.

![Graph D]

7. a) Give pseudo-code for the procedure bfs that performs a breath-first search of a graph $G$, starting from vertex $v$.

b) Give pseudo-code for the procedure dfs that performs a depth-first search of a graph $G$, starting from vertex $v$.

8. We wish to solve the $n$-Queens problem using the method of backtracking.

   a) Briefly describe what is meant by the $n$-Queen's Problem.
   b) Describe how it can be set up as a backtracking problem, i.e. describe the solution space and the state space tree.
   c) Give a mathematical formulation for the condition that two Queen's are attacking each other.
   d) Give pseudo-code for a backtracking algorithm for solving the $n$-Queens problem.
Design & Analysis of Algorithms Part of Qualifying Exam

Do 6 of the following 9 questions. Omit 3.

1. a) Using the definition of $\Theta$ show that $n^4 - 6n^3 + 4n^2 - 2 = \Theta(n^4)$.
   
b) Using limits and L'Hopital's rule show that $O(n) \in O(10^5)$.
   
c) Give an order of magnitude formula for each of the following, i.e. give the representative of the appropriate $\Theta$-class whose form is the simplest:
   
   (i) $1^5 + 2^5 + 3^5 + \ldots + n^5$
   
   (ii) $\frac{1}{3} + \frac{1}{4} + \ldots + \frac{1}{n}$
   
   (iii) $1 + n + n^2 + \ldots + n^k$
   
   (iv) \[ \left[ \frac{n!}{5} \right] \]
   
   d) Does $\log_2(2) + \log_2(3) + \ldots + \log_2(n) = \Theta(\log_2(n))$? Discuss.

2. Consider the recurrence relation:
   
   $$ t(n) = 6t(n-1) - 9t(n-2), \quad \forall n \geq 2, \quad t(0) = 0, \quad t(1) = 1. $$
   
   a) Obtain a closed-form formula for the associated generating function $g(x) = t(0) + t(1)x + \ldots + t(n)x^n + \ldots$
   
   b) Solve the above recurrence relation for $t(n)$ by finding the power series expansion of the formula for $g(x)$ you derived in a).

3. a) Give the tree of recursive calls to QuickSort, demonstrating the action of RearrangeAndPlace for each internal node of the tree, for the input list: 7 4 10 3 8 6 1 9 5 2
   
   b) Describe the tree of recursive calls for an input of size $n$ to QuickSort yielding the best case complexity $B(n)$, and give an order of magnitude formula for $B(n)$.
   
   c) Give the tree of recursive calls for an input of size $n$ to QuickSort yielding the worst case complexity $W(n)$, and give an order of magnitude formula for $W(n)$.

4. a) Give an $O(n^{\log_2 3})$ Divide-And-Conquer algorithm $\text{PolyMult}(P(x),Q(x),R(x))$ that computes the product polynomial $R(x)$ of two polynomials $P(x)$ and $Q(x)$ of size $n$.
   
   b) Give a complexity analysis of the algorithm you have given in a).
   
   c) Explain how a polynomial $P(x) = a_0 x^8 + \ldots + a_1 x + a_0$, and a quadratic $Q(x) = b_2 x^2 + b_1 x + b_0$ can be multiplied with the aid of repeated calls to PolyMult (without padding $Q(x)$ with zeros).
5. a) Assume the nodes of a binary tree are represented by dynamic variables having the structure:

\[ \text{TreeNode} = \text{record} \]

\[ \text{info: integer} \]
\[ \text{LeftChild: } \rightarrow \text{TreeNode} \]
\[ \text{RightChild: } \rightarrow \text{TreeNode} \]
\[ \text{end TreeNode} \]

Give pseudo-code for a recursive function \( \text{Leaf} (\text{Root}) \) that computes the number of leaves of the binary tree having root pointed to by \( \text{Root} \).

b) Give pseudo-code for the algorithm \( \text{DFT} (G) \) that performs the depth-first traversal of graph \( G \). Include pseudo-code for the recursive algorithm \( \text{DFS} (G,v) \) called by \( \text{DFT} \).

6. The polynomial \( P(x) \) of degree 5 assumes the same value as the quartic \( x^4 - x^3 + x^2 - x + 1 \) at \( x = -3, -2, -1, 2, 3 \). Further \( P(0) = 0 \).

Compute \( P(1) \).

7. \( \text{DFT}_i (P(x)) \) has coefficient array \( (0, -4+2i, -4, -2i) \). Find \( P(x) \), using the Fast Fourier Transform algorithm.

8. a) Explain what is meant by the Principle of Optimality in Dynamic Programming.

b) Show that the Principle of Optimality holds for shortest paths in graphs.

c) Consider the names: Abe, Betty, Joe, Karl, Mary, Tom, Zeus. Assuming that there is an equally likely chance of searching for any of the seven names above, give an optimal binary search tree for these names.

d) Using dynamic programming find an optimal binary search tree for the following probabilities:

\[
\begin{array}{c|cccc}
\text{keys} & 1 & 2 & 3 & 4 \\
\hline
p_i & .1 & .2 & .3 & .4 \\
\end{array}
\]

9. We wish to solve the \( n \)-Queens problem using the method of backtracking.

a) Briefly describe what is meant by the \( n \)-Queen's Problem.

b) Describe how it can be set up as a backtracking problem, i.e. describe the solution space and the state space tree.

c) Give pseudo-code for a backtracking algorithm for solving the \( n \)-Queens problem.
Problem 7 a) 
\[ G_1 = \]

Problem 7 b) 
\[ G_2 = 1 \]

Problem 8 b) 
\[ G_3 = \]
6. a) Give pseudo-code for quicksort. Do not include pseudo-code for the procedure pivot (also called partition).

b) Describe the tree of recursive calls for an input of size \( n \) to quicksort yielding the best case complexity \( B(n) \), and give an order of magnitude formula for \( B(n) \).

c) Give the tree of recursive calls for an input of size \( n \) to quicksort yielding the worst case complexity \( W(n) \), and give an order of magnitude formula for \( W(n) \).

d) Describe a modification to quicksort which gives an order of magnitude improvement in the computing time for a list of size \( n \) which is already sorted.

7. a) Using Kruskal's algorithm find a minimum cost spanning tree \( T \) in the weighted graph \( G_1 \) given on the next page (first graph). Mark the edges of \( T \) directly on the graph \( G_1 \).

b) Using Dijkstra's algorithm find a shortest path spanning tree \( T \) rooted at node 1 in the weighted digraph \( G_2 \) given on the next page (second graph), i.e., \( T \) contains a minimum length path from node 1 to all the other nodes where the length of a path is the sum of the weights of its edges. Mark the edges of \( T \) directly on the digraph \( G_2 \). Also label each node with the length of a shortest path from 1 to that node.

8. a) Give pseudo-code for BFS search which performs a breath-first search of a digraph \( G \).

b) Demonstrate the action of BFS for the graph \( G_3 \) given on the next page (third graph) by marking the breadth-first spanning tree \( T \) of \( G_3 \) and labelling the edges of \( T \) to indicate the order they are generated. Also demonstrate the action of the queue of visited but not explored nodes.
Design & Analysis of Algorithms Part of Qualifying Exam

Do 6 of the following 8 questions. Omit 2.

1. a) Give the definitions of $O$, $\Omega$, $\Theta$.
   b) Using the definition of $\Theta$ show that $n^3 - 50n^2 + 5 = \Theta(n^3)$.
   c) Show that $O(2^n) \subset O(3^n)$, i.e. $2^n = O(3^n)$ but $3^n \neq O(2^n)$.
   d) Does $1 + 2 + \ldots + n = \Theta(n)$? Discuss.

2. a) Give the pseudo-code for Linear Search whose input is a list $L[1..n]$ of size $n$ and a search element $X$, and whose output is the index of the first occurrence of $X$ or 0 if $X$ is not in the list.
   b) What are the best and worst case complexities of Linear Search? Explain.
   c) Assume that $L$ is a list of $n$ distinct elements, and that the probability that $X$ is found in the list is $3/4$. Also assume that, given that $X$ is in the list, there is an equally likely chance of $X$ occurring in any position. Determine the average case complexity of Linear Search.

3. a) Using the method of Divide-&-Conquer design an efficient algorithm called Fixed whose input is a sorted list $L[1..n]$ of integers, and whose output is an index $i$ such that $L[i] = i$ (fixed point), or 0 if no such index exists. Give pseudo-code for Fixed.
   b) What are the best and worst case complexities of the algorithm you have given in a). Explain.

4. a) Give pseudo-code for mergesort. Do not include pseudocode for procedure merge.
   b) Assume that the worst case complexity of procedure merge is $n$ obtain a recurrence relation (recurrence formula) for the worst case complexity $W(n)$ of Mergesort.
   c) Solve the recurrence relation you have given in b) to obtain an explicit formula for the worst case complexity of Mergesort.

5. a) Explain what is meant by the Principle of Optimality in Dynamic Programming.
   b) Show that the Principle of Optimality holds for shortest paths in a digraph.
   c) Give Floyd's algorithm (also called the all pairs shortest path algorithm) for finding the lengths of the shortest paths between every pair of nodes in digraph whose edges have been assigned a length (also called weight or cost). The input to Floyd is a matrix $L[1..n,1..n]$ giving the length of each edge, with $L[i,i] = 0$, $L[i,j] \geq 0$, and $L[i,j] = \infty$ if edge $(i,j)$ does not exist, and the output is a matrix $D[1..n,1..n]$ such that $D[i,j]$ is the length of a shortest path from $i$ to $j$. 
There are 8 questions. You have 2 hours to finish. Average time per question is 15 minutes.

1. a) Give the definition of \( f(n) = \Theta(g(n)) \).

   b) Give a linear function \( f \) and a quadratic function \( g \) such that \( f(n), g(n) \) are positive for all \( n \), and \( f(n) > g(n) \) for all \( n < 1000 \).

   c) Give a function \( f \) such that \( f = O(g(n)) \) but \( f \neq \Omega(g(n)) \).

2. Give the optimal binary search tree for \( (c_1, c_2, c_3, c_4) = (Abe, Joe, Pam, Zeke) \), where the probabilities of searching for \( c_1, c_2, c_3, c_4 \) are .1, .1, .1, .7, respectively. Explain your answer.

3. a) Give pseudo-code for Mergesort. Assume Merge has already been written (do NOT give pseudo-code for Merge).

   b) Give a recursion formula for the worst case complexity \( W(n) \) of Mergesort. Assume that the worst case complexity of Merge for a list of size \( n \) is \( n \).

   c) By solving the recursion formula you have given in b) show that the worst case complexity of Mergesort is \( O(n \log n) \).

4. a) Give a list of size \( n \) for which Quicksort takes the most computing time, i.e. whose computing time equals the worst case complexity \( W(n) \), and show that \( W(n) = \Theta(n^2) \).

   b) Quicksort can be made more efficient by applying the median rule, i.e. choosing the median of the first, middle, and last elements as the pivot element. Give the computing time of the list you have given in a) when the median rule is applied.

5. a) Find a least cost spanning tree in the weighted graph \( G \) above.

   b) Find a shortest path spanning tree in the weighted graph \( G \) above, i.e. a spanning tree \( T \) such that \( T \) contains a shortest (minimum cost) path from vertex 1 to each of the other vertices of \( G \).
6. a) Briefly explain what is meant by the Principle of Optimality in Dynamic Programming.

b) Show that the Principle of Optimality applies for the problem of finding a shortest path in a digraph.

c) Let $G$ be a digraph with vertex set $V = \{1, 2, \ldots, n\}$, and let $A^{(k)}(i, j)$ be the length of a shortest path from $i$ to $j$ whose internal vertices lie in the set $\{1, 2, \ldots, k\}$. Give a recursion formula for $A^{(k)}$.

7.

$$G = \begin{array}{c}
1 & 2 & 3 & 4 & 5 & 6 \\
\end{array}$$

a) Give the vertex 3-coloring of the graph above that results from backtracking.

b) Repeat a) for FIFO Branch-\&-Bound.

8. The quadratic $x^2 + x + 1$ interpolates the three points $(-1, 1)$, $(0, 1)$, and $(1, 3)$. Determine the cubic which interpolates these three points and the point $(5, 1)$ as well.