Analysis of Algorithms
Reviews

Reviews

1. Sorting
   - selection sort
   - insertion sort
   - bubble sort
   - quick sort
   - merge sort
   - heap sort
2. asymptotic notations
   - $\theta(f(n))$
   - $O(f(n))$
   - $O(f(n))$
   - $\Omega(f(n))$
   - $\omega(f(n))$
3. min-heaps and max-heaps
4. binary search trees
5. red-black trees
6. hashing
   - hashing with chaining
   - linear probing
7. divide and conquer
8. recursion trees
9. the master method
10. dynamic programming
    - top-down memoization
    - the matrix multiplication problem
    - the longest common subsequence
11. the greedy strategy
12. adjacency lists
13. adjacency matrices

14. breadth-first search
15. depth-first search
16. directed acyclic graphs
17. topological sorting
18. strongly connected components
19. minimum spanning trees
20. shortest paths
21. P vs NP
22. NP complete
23. NP hard
24. SAT
25. 3CNF SAT
26. the clique problem
27. the vertex cover problem
28. the Hamiltonian-cycle problem
29. the traveling-salesman problem
30. the subset-sum problem

Question 2

Write an algorithm in pseudo-code that takes as input an integer n and three arrays A, B, and C, each of n integers. The algorithm should return a tuple of three indices i, j, and k, such that A[i]+B[j] == C[k]. If there are multiple answers, then the algorithm can return any tuple. Prove the correctness of your algorithm, and analyze the asymptotic running time of the algorithm.

Question 3

Consider the following partitioning algorithm:

```
PARTITION(A,p,r)
    piv = A[p]
    i = p
    j = r
    while (i < j)
        while (i < j and A[j] > piv)
            j = j-1
        A[i] = A[j]
        i = i+1
        while (i < j and A[i] <= piv)
            i = i+1
        if (i < j)
            A[j] = A[i]
            j = j-1
    A[j] = piv;
    return j
```

- Prove that when PARTITION terminates, it returns a value `j` such that `p <= j <= r`, and every element of `A[p..j]` is less than or equal to `piv`, and every element of `A[j`1..r]+ is greater than `piv`.
- What value does PARTITION return when all elements in A[p..r] are the same?
- Modify the procedure so that it returns (p+r) div 2 when all elements in A[p..r] are the same.

Question 4

Let $f(n) = 56 \times n^3$ and $g(n) = 54 \times n^2$. Which of the following statements are true? (T or F. No explanation needed.)

1. $f(n) \in \omega(g(n))$
2. $f(n) \in \Omega(g(n))$
3. $f(n) \in \Theta(g(n))$
4. $f(n) \in o(g(n))$
5. $f(n) \in O(g(n))$

Question 5

Given a recurrence relation $T(n) = 7T(n/2) + n^3$, $T(1) = 1$.

1. Draw the first four levels of the recurrence tree.
2. Solve the recurrence to find a formula for $T(n)$.

Question 6

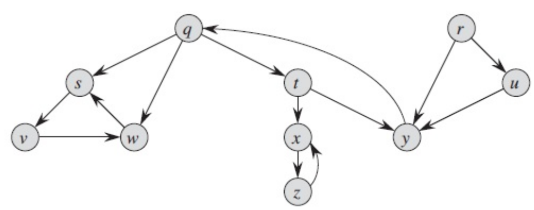
We can sort a given set of n numbers by first building a binary search tree containing these numbers (using TREE-INSERT repeatedly to insert the numbers one by one) and then printing the numbers by an inorder tree walk. What are the worst-case and best-case running times for this sorting algorithm?

Question 7

Write a top-down recursive algorithm for the #67th Euler Project problem (https://projecteuler.net/problem=67). Use dynamic programming to speed-up the algorithm.

Question 8

Consider the following directed graph:



1. Give an adjacency-matrix representation of the graph.
2. Give an adjacency-list representation of the graph, assuming that the vertices in adjacency lists are ordered alphabetically.
3. Show the distances that result from running breadth-first search on graph, using vertex u as the source.
4. Perform a DFS search on the graph. Fill in the parent, discovery time, and finishing time for each vertex.