Swarg

International Institute of Information Technology Hyderabad
Course: Algorithms

Mid Semester Examination I

September 7, 2018

Time: 90 mins. Max. Marks: **45**

There are nine questions in the paper, 5 marks each.
Answer all the questions in the answer sheet provided to you.

1. Make your own question that imaginatively tests one's understanding of `divide-and-conquer` algorithms (and solve it).

2. Fill in the following blanks:

   (a) The number of edges that are present in a forest of 23 trees on 101 vertices is ______.

   (b) Two complex numbers $(a + ib)$ and $(c + id)$ can be multiplied using just three (real) multiplications as follows: ______.

   (c) If $(\log n)!$ is polynomially bounded in n? ______.
   And a short proof of your answer is ______.

   (d) Give example of two sorting algorithms, say A and B, such that the worst-case asymptotic run-time of A is orders of magnitude faster than B for infinite values of input-size n and vice-versa (that is, the run-time of B is faster than A for infinite values of n). ______.

   (e) A Huffman encoding of ⟨`ebafcbcfbabadfacafabagaaffbfbafcafadfagaf`⟩ is ______.

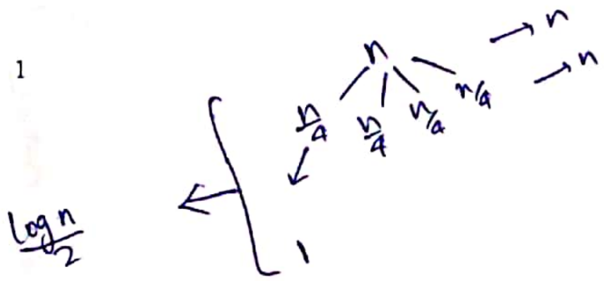
3. Prove that there must exist a problem that does not have a C program solving it.
`Hint:` Prove that the set of all computational problems with a natural number as input and Boolean output is *uncountable*. Prove that the set of all C programs is *countable*.

4. State and prove Fermat's Little Theorem and use it to *efficiently* evaluate: $a^{b^c} \bmod p$, for some prime p, where a, b, c are natural numbers.

5. Consider the problem of making change for n cents using the fewest number of coins. Assume that each coin's value is an integer.

   (a) Describe a greedy algorithm to make change if you have the denominations: 25 cents, 10 cents, 5 cents and 1 cent. Prove that your algorithm yields an optimal solution.

   (b) Suppose that the available coins are in the denominations that are powers of c, $c > 1$, i.e., the denominations are $c^0, c^1, \ldots, c^k$, $k \geq 1$. Show that the greedy algorithm always yields an optimal solution.



5. Suppose you have an arbitrary denomination of coins, say, $x_1, x_2, \ldots, x_k$, $x_i \geq 1$; prove that the greedy algorithm does not always work in this scenario.

6. *Convex Hull:* The convex hull of a set of points Q on a plane is the smallest convex polygon P for which each point in Q is either on the boundary or in its interior. To visualize a convex hull think of the points as nails sticking out of a wall; the convex hull is the shape formed by a rubber band stretched to encompass all the nails. For finding the convex hull of a given set of points, we can proceed as follows: At each step divide the set of points into two halves, recursively find the convex hull of each part and find a way to efficiently merge the two halves.

   Work out the details of this algorithm, derive a recurrence relation and solve it.

7. Consider each of the following words as a set of letters:

   {arid, dash, drain, heard, lost, nose, shun, slate, snare, thread, lid, roast}

   How does the output of greedy set cover (when ties are broken in favour of the word that appears first in the dictionary) when covering the set $X = \{a, d, e, h, i, l, n, o, r, s, t, u\}$ compare with an optimum solution for the above instance of minimum set-cover problem?

8. Illustrate how to multiply the two polynomials $3x^2 + 5x + 7$ and $2x + 1$ using the FFT. Specifically, choose an appropriate power of two, (recursively) find the FFT of the two sequences, multiply the results componentwise, and compute the inverse FFT to get the final result. Consider the problem of multiplying a polynomial with itself, that is squaring an n-degree polynomial. Do you think it is substantially easier than multiplying two arbitrary polynomials? Justify your answer. (`Hint:` Try reducing polynomial multiplication to squaring).

9. Make your own question that imaginatively tests one's understanding of `greedy` algorithms (and solve it).

ALL THE BEST

International Institute of Information Technology Hyderabad
Course: Algorithms

Mid Semester Examination II

October 15, 2018

Time: **90 mins.** Max. Marks: **45**

There are nine questions in the paper, 5 marks each.
Answer all the questions in the answer sheet provided to you.

1. Design and analyze an efficient algorithm to find the longest weighted path in an edge weighted tree. Does your algorithm work correctly if some of the edge weights are negative? Prove your answer.

2. You are given a stick and are asked to cut it at certain points P. The cost of cutting a stick of length L at any point is L. Design an algorithm to find a sequence of cuts such that the total cost is minimized. Prove the correctness of your algorithm. For example, given a stick of length 5 and asked to cut it at points $\{2, 4\}$ if you first cut at 4 and then at 2 the cost will be $5 + 4 = 9$, while if you make first cut at 2 and then at 4, total cost is $5 + 3 = 8$.

3. Consider the SUBSET-SUM problem: Given a set of integers, does the sum of some non-empty subset equal exactly zero? Show that SUBSET-SUM is NP-Complete. Assuming that the input consists only of numbers with absolute values being exact powers of 2, design and analyze a polynomial-time algorithm for SUBSET-SUM.

4. Compute the edit-distance between ALGORITHMS and ALTRUISTIC?

5. An old woman goes to market and a horse steps on her basket and crashes the eggs. The rider offers to pay for the damages and asks her how many eggs she had brought. She does not remember the exact number, but when she had taken them out five at a time, there were two eggs left. The same happened when she picked them out nine at a time, but when she took them seven at a time they came out three. What is the smallest number of eggs she could have had? What is the k^{th} smallest number of eggs she could have had?

$\sqrt{N}$, $a^b \geq N$

6. Design and analyze an efficient algorithm that checks if a given number is an *exact power*, that is, it is of the form a^b for some integers $a > 1$ and $b > 1$.

7. Geometrically/graphically solve the following non-convex optimization problem (similar to how we did linear programming in class geometrically).

$$\begin{aligned} \texttt{minimize } & x_1^2 + x_2^2 \\ \texttt{s.t. } 4 - x_1 - x_2^2 &\leq 0 \\ 3x_2 - x_1 &\leq 0 \\ -3x_2 - x_1 &\leq 0 \end{aligned}$$

8. Give the pseudocode and proof the correctness of probabilistic primality testing algorithm. Illustrate the working of the algorithm with a couple of examples.

9. Fill in the following blanks.

   (a) Is the longest increasing subsequence in a given sequence always unique? Justify your answer: ________.

   (b) The number of ways of distinctly parenthesizing a chain of 6 matrix multiplications is $\frac{1}{n}\binom{2n-2}{n-1}$.

   (c) Does "L is NP-Hard" imply "L is NP-Complete"? ×. Is the converse true? ✓.

   (d) The Longest Common Subsequence between ⟨s p e l l⟩ and ⟨h e l p⟩ is ____.

   (e) A recursive C function to compute the g.c.d. of two numbers is ________.

NP-Hard, NP-Complete, NP

ALL THE BEST

International Institute of Information Technology Hyderabad
Course: Algorithms

End Semester Examination

November 20, 2018

Time: 180 mins. **Max. Marks: 90**

There are eighteen questions in the paper, 5 marks each.
Answer all the questions in the answer sheet provided to you.

1. Recall the problem of MAX-CLIQUE — namely, to find the maximum sized complete sub-graph in a given graph G. Can you pose MAX-CLIQUE as a (possibly non-linear) mathematical programming problem?

2. Assuming that we augment the flow by 1 unit in each iteration of the Ford-Fulkerson algorithm give an example of a setting where the algorithm takes exponential (in the size of the network) iterations to terminate.

3. Design a *greedy* strategy to solve the minimum SET-COVER problem, with an approximation ratio of $O(\log n)$. Give a sufficient condition on the input instance so that your strategy finds that actual optimum. wst size fn?

4. Given a sorted array of distinct integers $A[1, \ldots, n]$, design an $O(\log n)$ algorithm to find out whether there is an index i for which $A[i] = i$. Bin search

5. The diameter of a tree $T = (V, E)$ is the largest of all shortest-path distances in the tree assuming all the edges to be undirected. Give an efficient algorithm to compute the diameter of a tree, and analyze running time of your algorithm.

6. Given 3-SAT is NP-complete, prove the NP-completeness of VERTEX-COVER.

7. Despite being NP-complete, show that when the input is restricted to *trees* the MINIMUM VERTEX COVER problem can be solved in *linear* time.

8. Show how to solve the maximum spanning tree problem using weighted *matroids*.

9. Very briely describe *ten* different algorithms that you know for sorting n numbers.

10. Given $p = 17$, $q = 23$, $e = 3$, what is the RSA public-key? What is the encryption of the message $m = 7$? What is the decrytpion of the ciphertext $c = 9$? $N = pq$, $\phi = (p-1)(q-1)$, $1 < e$ mod(e, ϕ) $ed \equiv 1 \bmod \phi$ $c = m^e \bmod N$

11. Prove using diagonalization that acceptance by a Turing machine (that is, given a TM M and its input w checking if M accepts w) is undecidable. $M = c^d \bmod N$

$18/3 = 6$

12. Solve the following cryptorithm: (each character needs to be mapped to a distinct digit between 0 and 9).

$$
\begin{array}{cccccc}
 & & & & & I \\
 & & L & I & K & E \\
 & & & T & H & E \\
+ & & S & I & L & K \\
\hline
 & S & H & I & R & T
\end{array}
$$

13. Solve the following system of linear congruences:

$$
\begin{aligned}
x &\equiv 1 \pmod{2}\\
x &\equiv 2 \pmod{3}\\
x &\equiv 3 \pmod{5}\\
x &\equiv 5 \pmod{7}\\
x &\equiv 7 \pmod{11}
\end{aligned}
$$

14. Show that *five* multiplications are sufficient to compute the square of a 2×2 matrix.

15. Give a modified Euclid's algorithm that runs in $O(\log(\max(a, b)))$ steps to compute the `l.c.m.` of two positive integers a and b. What is the worst-case input?

16. Find a longest increasing subsequence in $\langle$2 -2 1 -1 7 5 4 3 0 9 6 8$\rangle$.

17. Design, analyze and compare *five* different algorithms for computing the n^{th} Fibonacci number F_n, given n.

18. Write in detail about any *one* of the following:

(a) Quantum Teleportation

(b) The Fast Fourier Transform

(c) Randomized Algorithms

(d) The Simplex Algorithm

(e) Dual of a Linear Program (example MAX-FLOW and MIN-CUT)

(f) If a set $\mathcal{U}$ is countably infinite then its power set $2^{\mathcal{U}}$ is uncountable.

ALL THE BEST