CECS 528, Exam 1, Pink Version, Fall 2023, Dr. Ebert

Rules for Completing the Problems

NO NOTES, BOOKS, ELECTRONIC DEVICES, OR INTERPERSONAL COMMUNICATION allowed when solving these problems. Make sure all these items are put away BEFORE looking at the problems. FAILURE TO ABIDE BY THESE RULES MAY RESULT IN A FINAL COURSE GRADE OF F.

Directions

Choose up to **six problems** to solve. Clearly mark each problem you want graded by placing an X or check mark in the appropriate box in the Grade(?) row of the table below. **If you don't mark any problems for us to grade or mark 7 or more problems, then we will record grades for the 6 that received the *fewest* points.**

Problem	1	2	3	4	5	6
Grade?						
Result						

Your Full Name:

Your Class ID:

1. Solve each of the following problems. Note: correctly solving these problems counts for passing LO1.

   a. Compute the multiplicative inverse of 25 mod 36. (10 pts)

$36 - 25(1) = 11$

$25 - 11(2) = 3$

$11 - 3(3) = 2$

$3 - 2 = 1$

$3 + 2(-1) = 1$

$3 + [11 - 3(3)](-1) = 1$

$3(4) - 11 = 1$

$[25 - 11(2)](4) - 11 = 1$

$25(4) - 11(9) = 1$

$25(4) - [36 - 25(1)](9) = 1$

$25(13) - 36(9) = 1$

$\therefore$ 13 = multiplicative inverse of 25 mod 36

   b. For the Strassen-Solovay primality test is $a = 11$ an accomplice or witness to the fact that $n = 15$ is not prime? Show all work. (15 pts)

$a^{\frac{n-1}{2}} = 11^7$

$\therefore a^{\frac{n-1}{2}} \equiv \left(\frac{a}{n}\right) \bmod n$

$11^7 \equiv \left(\frac{11}{15}\right) \bmod 15$

$(-4)^7 \bmod 15 \Rightarrow \cancel{(-4)^6}^{\,1}(-4) \bmod 15$

$\Rightarrow -4 \bmod 15$

$\Rightarrow 11$

$\frac{11}{15} \bmod 15$

$\Rightarrow \frac{-4}{15} \bmod 15$

$\Rightarrow (-1)\left[\cancel{\frac{2}{15} \times \frac{2}{15}}^{\,1}\right] \bmod 15$

$\Rightarrow (-1)$

$\therefore 11 \not\equiv -1 \bmod 15$

$\therefore n = 15$ is not a prime

2. Solve each of the following problems. Note: correctly solving these problems counts for passing LO2.

   a. Use the Master Theorem to determine the growth of $T(n)$ if it satisfies the recurrence $T(n) = 56T(n/4) + n^3$. (10 pts)

$n^{\log_b a} = n^{\log_4 56}$ $\because$ $\log_4 56 < 3$

$\therefore$ $f(n) = \Omega(n^{\log_b a + \epsilon})$

$\therefore$ $\epsilon = 3 - \log_4 56$

$\therefore$ By case 3

$T(n) = \Theta(f(n))$

$= \Theta(n^3)$

   b. Use the substitution method to prove that, if $T(n)$ satisfies

$$T(n) = 4T(n/2) + n^2 \log n,$$

   Then $T(n) = \Omega(n^2 \log^2 n)$. (15 pts)

$T(k) \geq ck^2\log^2 k$

$T(n) = 4T(n/2) + n^2\log n$

$= \cancel{4}\left[\frac{cn^2}{\cancel{4}}\log^2\frac{n}{2}\right] + n^2\log n \geq cn^2\log^2 n$

$= c\cancel{n^2}(\log n - \log 2)^2 + \cancel{n^2}\log n \geq c\cancel{n^2}\log^2 n$

$= c(\log n - 1)^2 + \log n \geq c\log^2 n$

$= c(\log^2 n - 2\log n + 1) + \log n \geq c\log^2 n$

$= \cancel{c\log^2 n} - 2c\log n + c + \log n \geq \cancel{c\log^2 n}$

$= \log n \geq 2c\log n - c$

$= \frac{\log n}{2\log n - 1} \geq c$

let $c = 1/2$

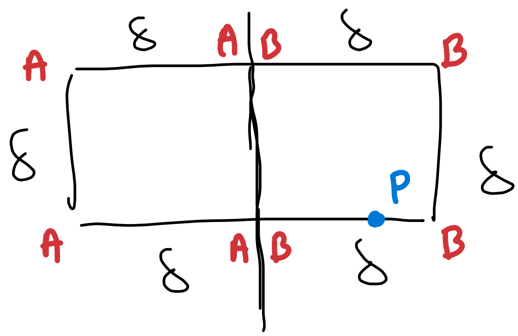
$\frac{\log n}{2\log n - 1} \geq \frac{1}{2}$

$2\cancel{\log n} \geq \cancel{2\log n} - 1$

$\therefore$ $0 \geq -1$ $c = 1/2$

3. Solve each of the following problems. Note: correctly solving these problems counts for passing LO3.

   a. Recall the combine step of the `Minimum Distance Pair (MDP)` algorithm where, for each point P in the δ-strip, there is a $2\delta \times \delta$ rectangle whose bottom side contains P and is bisected by the vertical line that divides the points into left and right subsets. Explain why there can be at most 7 other points (from the problem instance) in this rectangle. (12 pts) Any 2 points in the square are δ length away from each other. At most 4 points can fit in the $\delta \times \delta$ square. With one point being P itself, it is then compared with 7 points at max.



   b. Recall that the `Minimum Positive Subsequence Sum (MPSS)` problem admits a divide-and-conquer algorithm that, on input integer array a, requires computing the mpss of any subarray of a that contains both $a[n/2-1]$ and $a[n/2]$ (the end of a_{left} and the beginning of a_{right}). For

$$a = 48, -37, 29, -33, 51, -64, 46, -34, 45, -36$$

   Provide the two sorted arrays a and b from which the minimum positive sum $a[i]+b[j]$ represents the desired mpss, for some i in the index range of a and some j within the index range of b. Also, demonstrate how the minimum positive sum $a[i]+b[j]$ may be computed in $O(n)$ steps. (13 pts)

48 -37 29 -33 51 | -64 46 -34 45 -36

Left sum = 51, 18, 47, 10, 58 ; sort = 10, 18, 47, 51, 58

Right sum = -64, -18, -52, -7, -43 ; sort = -64, -52, -43, -18, -7

l	r	sum
0	4	3 → MPSS
0	3	-8
1	3	0
2	3	29
2	2	4
2	1	-5
3	1	-1
4	1	6
4	0	-6

∴ MPSS = 3

It will take O(n) steps as the pointers will be moving n/2 from both sides

4. In this problem we assume that multiplication of an m-bit number with an n-bit number results in a product having $m+n$ bits, and that requires $O(mn)$ steps to compute. Using these assumptions, determine the worst-case running time for the following code that computes x^y, where we assume x is an m-bit number and y is an n-bit number. (25 pts)

```
prod = x;

for(i=1; i < y; i++)
    prod = prod*x;

return prod;
```

Since we are multiplying x with x instead of $(m+n)$ it will be $m+m$

$(m+m)$, $(2m+m)$, $(3m+m)$ & so on.

Instead of (mn) it will be $(m \times m)$

(m^2), $(2m^2)$ $(3m^2)$

$$\sum_{i=1}^{n-1} im^2 \Rightarrow m^2 \sum_{i=1}^{n-1} i$$

$$\Rightarrow O\left(m^2 \frac{(n-1)n}{2}\right)$$

$$\Rightarrow O(m^2 n^2)$$

5. Recall the Master Equation

$$T(n) = \Theta(n^{\log_b a}) + \sum_{j=0}^{\log_b n - 1} a^j f(n/b^j).$$

Assuming n is a power of b, suppose that $f(n) = \Omega(n^{\log_b a+\epsilon})$ for some constant $\epsilon > 0$, and that, for all $n \geq 1$, $af(n/b) \leq cf(n)$ for some constant positive $c < 1$, then prove that $T(n) = \Theta(f(n))$. (25 pts) $\because n = b^i$

$\therefore T(n) = aT(n/b) + f(n)$

$\because f(n) = \Omega(n^{\log_b a + \epsilon})$

Since $f(n)$ is equal to the first term of the sum that adds to $g(n)$ & all terms are nonnegative, we have $g(n) = \Omega f(n)$.

$\therefore$ acc. to case 3:

$$a^j f\left(\frac{n}{b^j}\right) \leq c^j f(n) \sum_{j=0}^{\log_b n - 1} c^j \leq \left(\frac{1-c^{\log_b n}}{1-c}\right) f(n)$$

$\therefore g(n) \leq f(n)$

$\therefore g(n) = O(f(n))$

$\therefore g(n) = \theta(f(n))$

6. The Hadamard matrices H_0, H_1, $H_2 \ldots$ are recursively defined as follows. H_0 is the 1×1 matrix [1], and, for $k \geq 1$, H_k is the $2^k \times 2^k$ matrix

$$H_k = \left(\begin{array}{c|c} H_{k-1} & H_{k-1} \\ \hline H_{k-1} & -H_{k-1} \end{array} \right).$$

Describe an algorithm that computes the matrix-vector product $H_k v$ using $O(n \log n)$ operations, where v is a column vector of length $n = 2^k$. Assume that all the numbers involved are small enough so that basic arithmetic operations like addition and multiplication take unit time. Note: credit will not be awarded to descriptions that are ambiguous, make incorrect assumptions/conclusions, and/or do not achieve the desired bound on operations. (25 points)

Let $T(n)$ denote the no. of operations required to multiply H_k with a vector v of length $n=2^k$. Let V_u denote the upper $n/2 = 2^{k-1}$ entries of V and V_l denotes the lower $n/2 = 2^{k-1}$ entries of V.

Now compute

W_1 = H_(k-1) x V_u & W_2 = H_(k-1) x V_l

Then the upper half of H_k x V equals W_1 + W_2 while the lower half of H_k x V equals W_1 - W_2.

The running time of the algorithm is $T(n) = 2T(n/2) + O(n)$ where $2T(n/2)$ is from the two subproblems W_1 & W_2. $O(n)$ is from adding & subtracting.

$\therefore T(n) = O(n \log n)$.