WEEK 11 : CS275 RECITATION EXERCISES - SOLUTION

For each question, read **each word** with the greatest care and **without hurrying**. If you have doubts about what is asked, **go back** to the wording of the question until the meaning of the question is clear. Then try to find an answer.

1. Solving the recursive definition of the complexity of an algorithm

"Master theorem".
If the complexity of an algorithm, when given n input elements, is $f(n)$ and the function f verifies the recurrence relation

$$f(n) = af(n/b) + cn^d, \tag{1.1}$$

then,

a) if $a > b^d$, then $f(n) = O\left(n^{\log_b a}\right)$.
b) if $a < b^d$, then $f(n) = O\left(n^d\right)$.
c) if $a = b^d$, then $f(n) = O\left(n^d \log_b(n)\right)$.

See also Section 6.3 of [1].

Proof. We only consider the case in which n is a power of b. Let $L = \log_b(n) \in \mathbb{N}$, so that $n = b^L$.

$$\begin{array}{rcll} f(n) & = & af\left(\frac{b^L}{b}\right) + cb^{Ld} & \text{By (1.1) and def of } L \\ & = & a\left(af\left(b^{L-2}\right) + cb^{(L-1)d}\right) + cb^{Ld} & \text{Eq. (1.1) again.} \\ & = & a^2 f\left(b^{L-3}\right) + cab^{(L-1)d} + cb^{Ld} & \text{and again} \\ & = & a^3 f\left(b^{L-4}\right) + ca^2 b^{(L-2)d} + cab^{(L-1)d} + cb^{Ld} & \vdots \\ \vdots & \vdots & & \vdots \\ & = & a^L f(1) + ca^{L-1}b^d + ca^{L-2}b^{2d} + \ldots + cab^{(L-1)d} + cb^{Ld}. & \end{array} \tag{1.2}$$

Which of the term(s) in this expression dominate the others?

a) If $a > b^d$, then a^L dominates $a^{L-i}b^{id}$, for all $1 \leq i \leq L$, so that $f(n) \in O\left(a^L\right)$. Since $a^L = a^{\log_b(n)} = b^{\log_b(n)\log_b(a)} = n^{\log_b(a)}$, one has $f(n) \in O\left(n^{\log_b(a)}\right)$.
b) If $a < b^d$, then b^{Ld} dominates $a^{L-i}b^{id}$, for all $0 \leq i < L$, so that $f(n) \in O\left(b^{Ld}\right) = O\left(n^d\right)$.
c) If $a = b^d$, then Eq. (1.2) becomes

$$\begin{array}{rcl} f(n) & = & b^{Ld} f(1) + cb^{Ld} + cb^{Ld} + \ldots + cb^{Ld} \\ & = & n^d\left(Lc + f(1)\right), \end{array}$$

so that $f(n) \in O\left(n^d \log_b(n)\right)$.

2. Exercises

Exercise 1. Sort these lists of numbers using the merge sort algorithm:

a) 5, 5, 10, 8, 12, 9, 2, 9, 7, 3, 7, 5, 5, 7, 4, 11.
Solution:

1) (5,5,8,10,9,12,2,9),(3,7,5,7,5,7,4,11) — Split list.
2) (5,5,8,10),(9,12,2,9),(3,7,5,7),(5,7,4,11) — Split lists.
3) (5,5),(8,10),(9,12),(2,9),(3,7),(5,7),(5,7),(4,11) — Split lists.
4) (5),(5),(8),(10),(9),(12),(2),(9),(3),(7),(5),(7),(5),(7),(4),(11) — Split lists.
5) (5,5),(8,10),(9,12),(2,9),(3,7),(5,7),(5,7),(4,11) — Sort lists of length 1 and merge them.

6) (5,5,8,10),(2,9,9,12),(3,5,7,7),(4,5,7,11) Merge sorted lists of length 2.
7) (2,5,5,8,9,9,10,12), (3,4,5,5,7,7,7,11) Merge sorted lists of length 4.
8) (2,3,4,5,5,5,5,7,7,7,8,9,9,10,11,12). Merge sorted lists of length 8.

b) 4, 3, 11, 0, 12, 8, 7, 10, 8, 2, 3.
Solution:
1) (4, 3, 11, 0, 12, 8), (7, 10, 8, 2, 3) Split list.
2) (4, 3, 11), (0, 12, 8), (7, 10, 8), (2, 3) Split lists.
3) (4, 3), (11), (0, 12), (8), (7, 10), (8), (2), (3) Split lists.
4) (4),(3), (11), (0), (12), (8), (7), (10), (8), (2), (3) Split lists and sort lists of length 1.
5) (3,4),(11), (0, 12), (7,10), (8), (2,3) Sort lists of length 1 and merge (if possible).
6) (3, 4, 11), (0,8,12),(7, 8, 10), (2, 3) Merge sorted lists.
7) (0, 3, 4, 8, 11, 12), (2,3,7,8,10) Merge sorted lists.
8) (0, 2, 3, 3, 4, 7, 8, 8, 10, 11, 12). Merge sorted lists.

Solution: Should not be needed.

Exercise 2. A person A chooses a number in $\{0 \ldots 2^n - 1\}$. Person B will guess the value of the number by asking questions, to which A will answer (truthfully) by "yes" or "no."

a) Find a way of formulating questions, so that B can guess the number as quickly as possible.
Solution: Question1: Is $x \geq 2^{n-1}$? Question 2: If answer to prev question was "yes": is $x \geq 2^{n-1} + 2^{n-2}$? Else : is $x \geq 2^{n-2}$ etc, each question reducing by half the set in which x can lie.
Simpler formulation: Write x in base 2: $x = x_0 + 2x_1 + 4x_2 + \ldots + 2^{n-1}x_{n-1}$. Question i is then: Is $x_{i-1} = 0$? for $1 \leq i \leq n$.

b) How many questions are needed? Call this $f(n)$. .... $f(n) = n$.
c) Find a recurrence relation expressing $f(n)$ as a function of $f(n-1)$. ...... $f(n) = f(n-1) + 1$.

Exercise 3. Solve Exercise 4 p. 423 of [1]. Solve these recurrence relations together with the initial conditions given

a) $a_n = a_{n-1} + 6a_{n-2}$ for $n \geq 2$, $a_0 = 3$, $a_1 = 6$
Solution: Characteristic polynomial: $r^2 - r - 6 = 0$, discriminant 25 roots $r_1 = \frac{1+5}{2} = 3 \neq r_2 = \frac{1-5}{2} = -2$, $\alpha_1 = 2.4$ and $\alpha_2 = 0.6$, so that $a_n = 2.5 \cdot 3^n + 0.6 \cdot (-2)^n$.

b) $a_n = 7a_{n-1} - 10a_{n-2}$ for $n \geq 2$, $a_0 = 2$, $a_1 = 1$.
Solution: Roots: $r_1 = 5 \neq r_2 = 2$, $\alpha_1 = -1$ and $\alpha_2 = 3$, so that $a_n = -5^n + 3 \cdot 2^n$.

c) $a_n = 6a_{n-1} - 8a_{n-2}$ for $n \geq 2$, $a_0 = 4$, $a_1 = 10$
Solution: Roots: $r_1 = 4 \neq r_2 = 2$, $\alpha_1 = 1$ and $\alpha_2 = 3$, so that $a_n = 4 + 3 \cdot 2^n$.

d) $a_n = 2a_{n-1} - a_{n-2}$ for $n \geq 2$, $a_0 = 4$, $a_1 = 1$
Solution: Roots: $r_1 = 1 = r_2$, $\alpha = 4$ and $\beta = -3$, so that $a_n = 4 - 3n$.

e) $a_n = a_{n-2}$ for $n \geq 2$, $a_0 = 5$, $a_1 = -1$
Solution: $a_{2n} = 5$, $a_{2n+1} = -1$.

$a_n = -6a_{n-1} - 9a_{n-2}$ for $n \geq 2$, $a_0 = 3$, $a_1 = -3$
Solution: Roots: $r_1 = -3 = r_2$, $\alpha = 3$ and $\beta = -2$, so that $a_n = (-3)^n (3 - 2n)$.

a) $a_{n+2} = -4a_{n+1} + 5a_n$ for $n \geq 0$, $a_0 = 2$, $a_1 = 8$
Solution: Roots: $r_1 = 1 \neq r_2 = -5$, $\alpha_1 = 3$ and $\alpha_2 = -1$, so that $a_n = 3 - (-5)^n$.

Exercise 4. Solve Exercise 7 p. 423 of [1]. In how many ways can a $2 \times n$ rectangular board be tiled using 1×2 and 2×2 pieces?

Solution: Let a_n be the number of ways to tile a $2 \times n$ board. There is just $a_1 = 1$ way of tiling a 2×1. There are $a_2 = 3$ ways to tile a 2×2 board (Fig. 4). A $2 \times n$ tiled board can be obtained either by

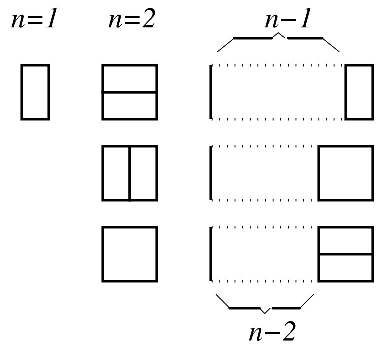

FIGURE 3.1. Exercise 4.

- appending a vertical 2×1 piece to a $2 \times (n-1)$ board tiled or
- by appending two horizontal 1×2 pieces to a $2 \times (n-2)$ tiled board, or
- by appending a 2×2 piece to a $2 \times (n-2)$ tiled board.

These three cases disjoint and are the only ways of tiling a $2 \times n$ board. One thus has $a_n = a_{n-1} + a_{n-2} + a_{n-2} = a_{n_1} + 2a_{n-2}$.
The characteristic equation is $r^2 - r - 2 = 0$, which has roots 2 and -1, so that $a_n = \frac{2}{3}2^n + \frac{1}{3}(-1)^n$; make sure you solve with initial conditions x_1 and x_2, not x_0 and x_1.

Exercise 5. Solve Exercise 14 p. 434 of [1]. Suppose that there are $n = 2^k$ teams in an elimination tournament, where there are $n/2$ games in the first round, with the $n/2 = 2^{k-1}$ winners playing in the second round, and so on. Develop a recurrence relation for the number of rounds in the tournament.

Solution: With one more round, one can accomodate twice the number of teams: $f(2n) = f(n) + 1$. $f(1) = 0$, since no round is needed if one team only is present.

Exercise 6. Solve Exercise 15 p. 434 of [1]. How many rounds are there in the elimination tournament described in Exercise 14, when there are 32 teams?

Solution: $f(1) = 0$, $f(2) = 1$, $f(4) = 2$, $f(8) = 3$, $f(16) = 4$, $f(32) = 5$.
The recursion relation is $f(n) = f(n/2) + 1$, which, corresponds to $a = 1$, $b = 2$, $c = 1$ and $d = 0$ in Eq. (1.1) above. Since $a = b^d$, the complexity is $f(n) \in O(\log_2(n))$.

REFERENCES

[1] K. H. Rosen. *Discrete Mathematics and Its Applications*. Mc Graw Hill, 5 edition, 2003.