WEEK 11 : CS275 RECITATION EXERCISES

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 list of numbers using the merge sort algorithm:

a) 5, 5, 10, 8, 12, 9, 2, 9, 7, 3, 7, 5, 5, 7, 4, 11.
b) 4, 3, 11, 0, 12, 8, 7, 10, 8, 2, 3.

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

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 $n \geq 2$, $a_0 = 3$, $a_1 = 6$
b) $a_n = 7a_{n-1} - 10a_{n-2}$ for $n \geq 2$, $a_0 = 2$, $a_1 = 1$.
c) $a_n = 6a_{n-1} - 8a_{n-2}$ for $n \geq 2$, $a_0 = 4$, $a_1 = 10$
d) $a_n = 2a_{n-1} - a_{n-2}$ for $n \geq 2$, $a_0 = 4$, $a_1 = 1$
e) $a_n = a_{n-2}$ for $n \geq 2$, $a_0 = 5$, $a_1 = -1$
f) $a_n = -6a_{n-1} - 9a_{n-2}$ for $n \geq 2$, $a_0 = 3$, $a_1 = -3$
g) $a_{n+2} = -4a_{n+1} + 5a_n$ for $n \geq 0$, $a_0 = 2$, $a_1 = 8$

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?

Exercise 5. Almost like Exercise 1 p. 433 of [1]. How many comparisons are needed for a binary search in a set of 64 elements? What about in a sorted set?

Exercise 6. 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.

Exercise 7. 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?

References

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