

**CS161 SU24: Homework 2 (Due July 10 11:00am)****❖ Problem 1**

Use the master theorem to solve the following recurrence relations.

1.  $T(n) = 64T(n/4) + 16n^3 \log^2 n$ .

2.  $h(n) = 1000h(n/10) + 2n^5 \log n$ .

3.  $P(n) = 15P(n/3) + 40n^{2.5}$ .

4.  $W(x) = 32W(x/2) + 5x^3$ .

5.  $J(y) = 3J(3y/4) + 2y^3$ .

6.  $R(m) = 4R(m/3) + 3\sqrt{m}$

## ❖ Problem 2

We want to play a round-robin tennis tournament with  $n$  players where  $n$  is a power of 2. A round-robin tournament is a tournament where every player plays every other player exactly one time. For our tennis tournament, we want every player to play one game per day, for a total of  $n - 1$  days.

1. Design an algorithm that generates a schedule for our tournament. You may want to have an empty  $n$  by  $n - 1$  grid which each recursive call has access to. Each row corresponds to a player  $p$ , and each column is a day  $d$ . Each cell stores  $p$ 's opponent on day  $d$ . See the figure below for an example.

(Hint: Use divide and conquer in the following way. First divide the players into two equal groups and let them play within each group for the first  $n/2 - 1$  days. Then, schedule the games between the two groups for the other  $n/2$  days. Try it by hand to see how the algorithm would work in general.)

2. Write down the recurrence relation that captures the runtime of this algorithm.
3. Solve the recurrence relation you wrote above.

2	3	4
1	4	3
4	1	2
3	2	1

Figure 1: An example schedule of a four player round robin tournament. Each row is a player, and each column corresponds to a different day, representing the opponent of that player. For example, player 3 plays against player 1 on day 2.

## ❖ Leetcode

Here is a sample of some Leetcode problems related to divide and conquer that I think can be solved/understood with what we have covered so far. As mentioned at the beginning of class, set a timer to try solving these on your own, then once the timer is up check a solution and try to understand why that works.

For any solution you write, try to come up with the recurrence relation and solve it to get your final run time.

- Majority element [Easy]

<https://leetcode.com/problems/majority-element/description/>

- Number of 1 bits [Easy]

<https://leetcode.com/problems/number-of-1-bits/description/>

- $k$ -th largest element in an array [Medium]

<https://leetcode.com/problems/kth-largest-element-in-an-array/description/>

- Construct Quad Tree [Medium]

<https://leetcode.com/problems/construct-quad-tree/description/>

- The Skyline Problem [Hard]

<https://leetcode.com/problems/the-skyline-problem/description/>