Week 6 Day 1 MIDTERM REVIEW

Prep:

- Midterm Review Problems posted
- Solutions to problems are also posted.
- Review Quiz answers.

Integral Scheduling: (Greedy Algorithm Slides)

- Greedy Algorithm: Solving using the simplest algorithm to implement.
- Doesn't guarantee optimal runtime but can be useful.

Greedy Algorithm for Heurstics:

Option 1: Select earliest starting task

Option 2: Select the Shortest Available Task

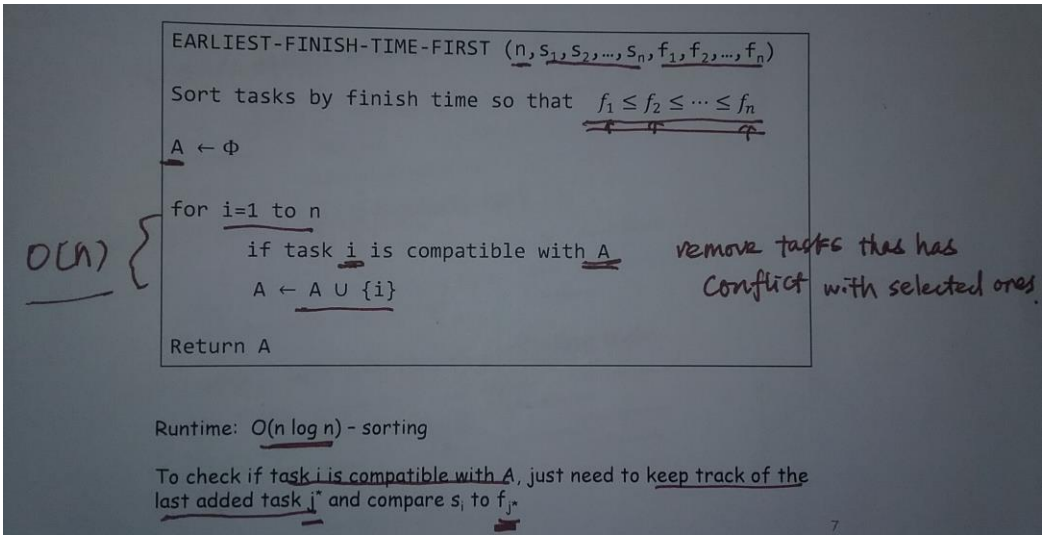
Option 3: Schedule task with fewest conflicting tasks

- **All 3 of these greedy options don't guarantee an optimal solution**
- Come up with examples that fail with these criteria.

Option 4: Earliest Finishing Time

Earliest Finish Time First (Algorithm Pseudo-Code)

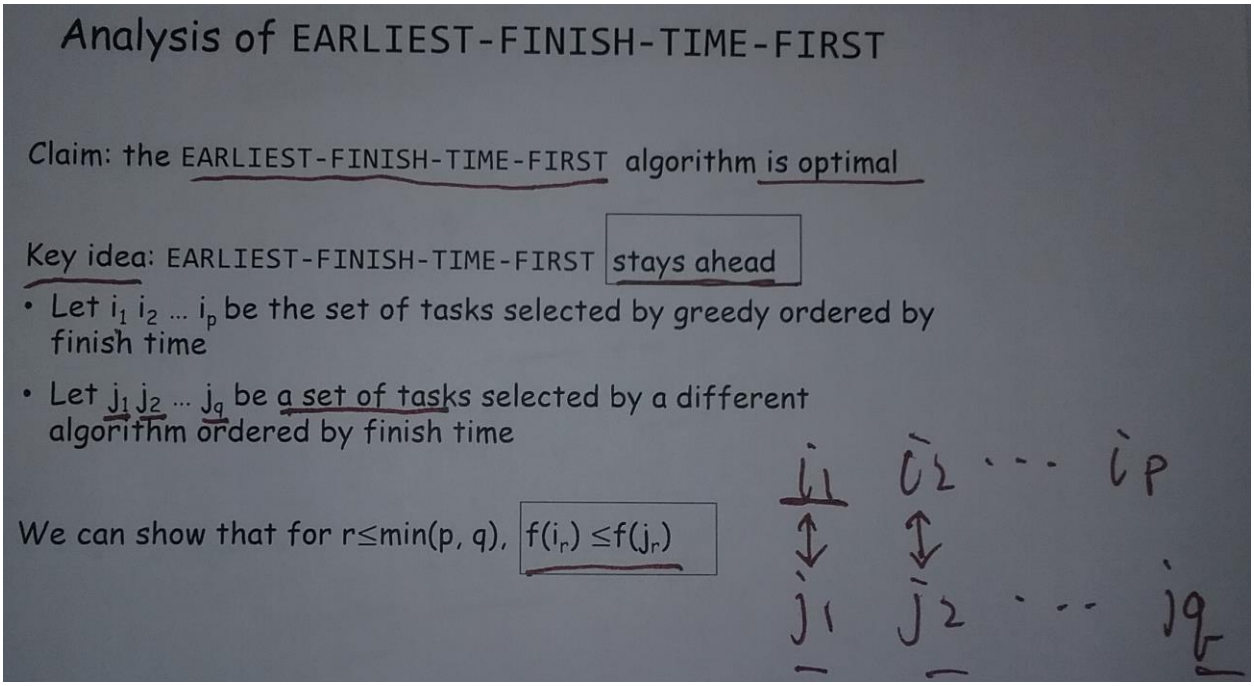
- **Sort Tasks** in finish time order.



Runtime Speed: O(n log n) – Sorting

To check for compatibility: Compare Current last finish time and new start time.

EARLIEST-FINISH-TIME-FIRST Optimal (**Analysis Picture*) EFTF**



Proof by Induction (See Stay Ahead Lecture Slides*)

J_{R+1} Must be compatible with i_{R+1} because of the definition of how the greedy algorithm chooses the next element. Therefore, the finish time for the K+1 step is no later than the J_{r+1}.

Completing the Proof:

The EFTF solution has to have an equal number of tasks as the optimal solution q.

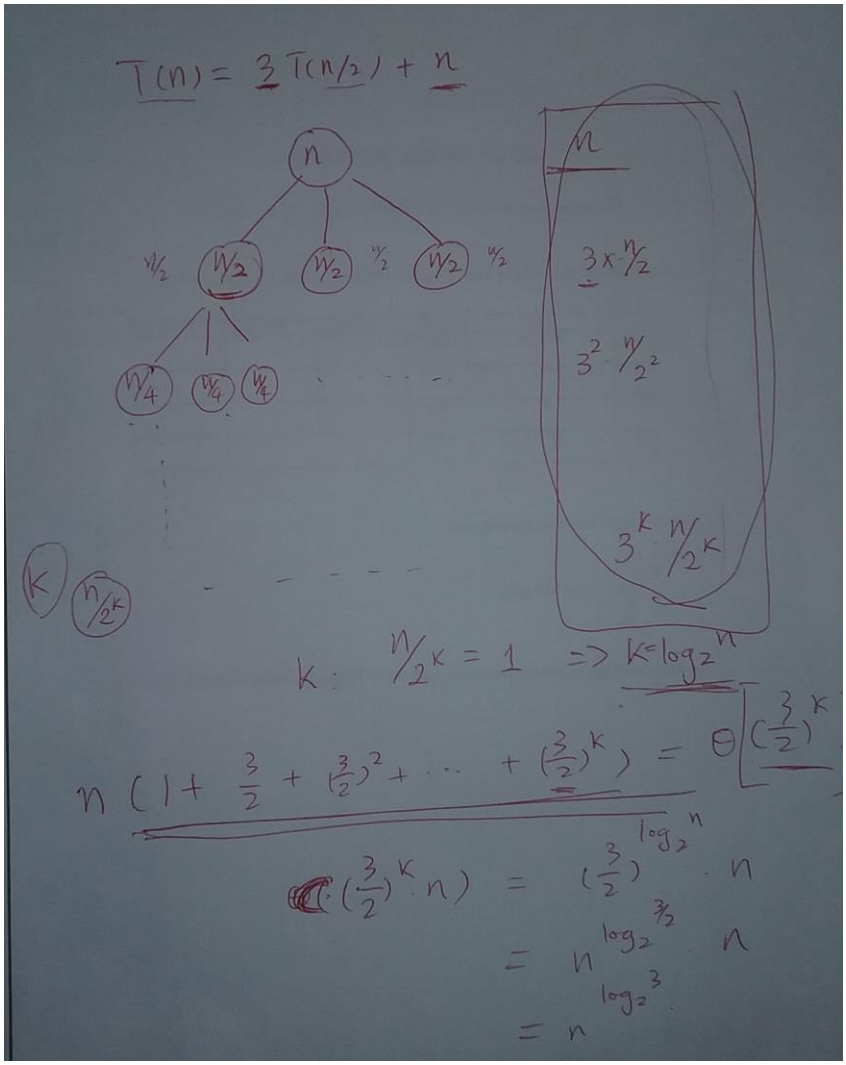
So p must be equal to q because if it's less than q than q is not the optimal solution. p = q. p cannot be less than q.

Midterm Review:

Asymptotic Runtime Analysis:

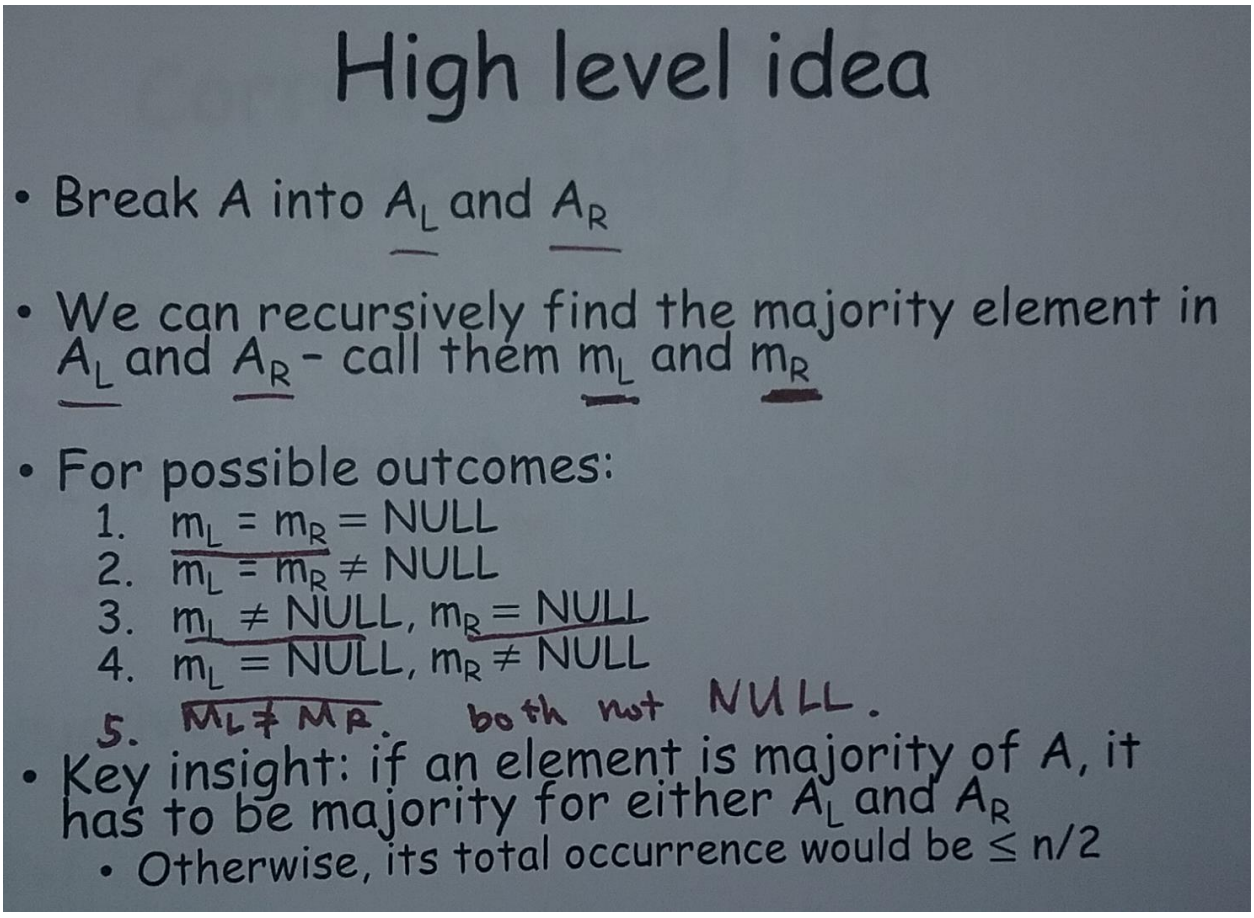
- Big O(<=), Theta (=) Big Omega (>=)
- Email of Asymptotic order strategies.
- T(n) = T(n-1) + T(n-2) <= 2T(n-1)
- Rule of Thumb.
- Limit of Ratio. (Lim as n → Infinity of: f(n) / g(n)
- Convert Pseudo-Code to Runtime.
- Recursion Tree for Solving Recurrence relations.
- Master Theorem.
- Geometric Series

Recursion Tree Example: (Picture)



Divide and Conquer:

- Break down into small sub-problems
- Solve them via recursion
- Combine solutions of sub-problems to get the original problem
- Example: merge sort (n log n) break into half over and over. Spend linear time to combine them back.
- Binary search (log n) Don't need to recombine.
- Merge Sort Recurrence Relation: (T(n) = T(n/2) + C)
- **Majority Element Problem: (from Midterm Review)**
- *High Level Idea Picture*

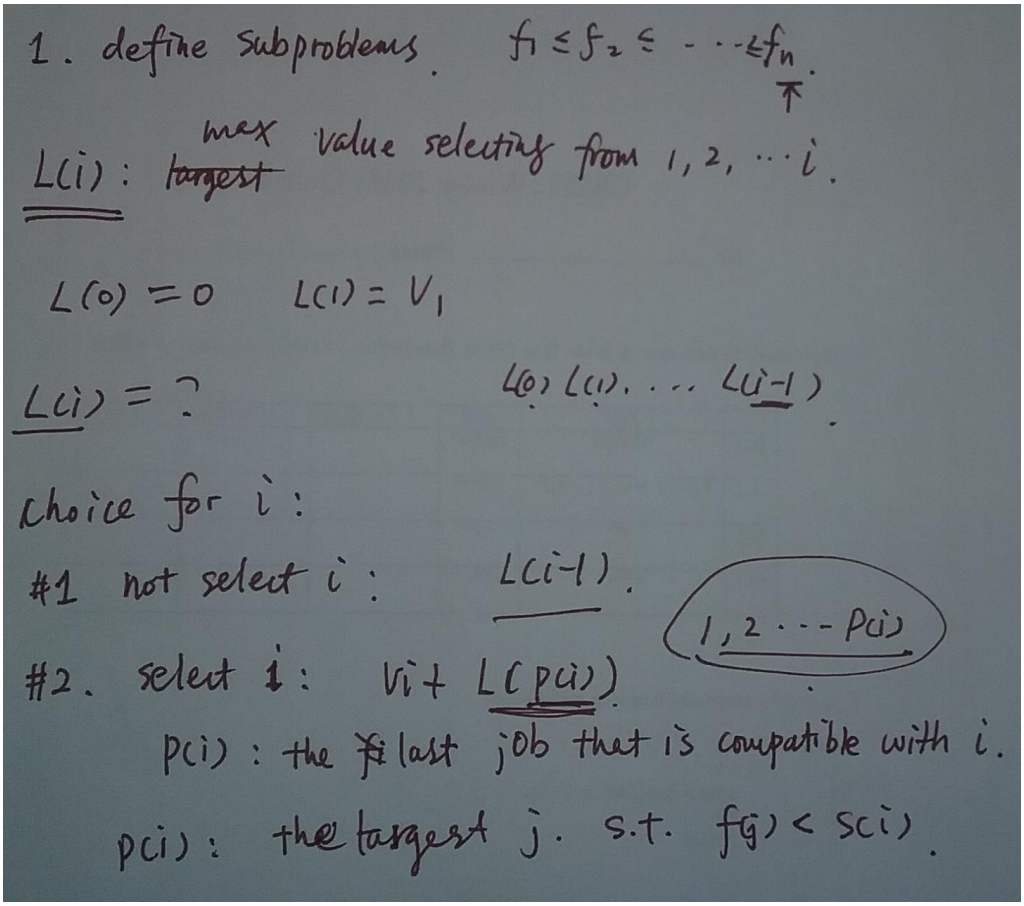


Proof by Induction:

- Base Case
- Inductive Assumption: Assume correct up to k.
- Inductive Strep: Consider k+1

Greedy Interval with Weighted Scheduling and Dynamic Programming:

Dynamic Programming: (See Picture*)



- 1. Define Subproblems to solve.
- 2. Give recurrence relation for the Subproblems. Provide Base Case.
- 3. Give runtime of algorithm and explain.

Next Time:

- Review for **Midterm** on Thursday
- More Officer Hours will be available for Wednesday (check Email and Canvas for Dates)
- No Recitation Wednesday.
- **Continue Working on Assignment 2**

End of Week 6 Day 1 MIDTERM REVIEW Notes

~Information composed by Notetaker Scott Russell for CS 325 **DAS** student