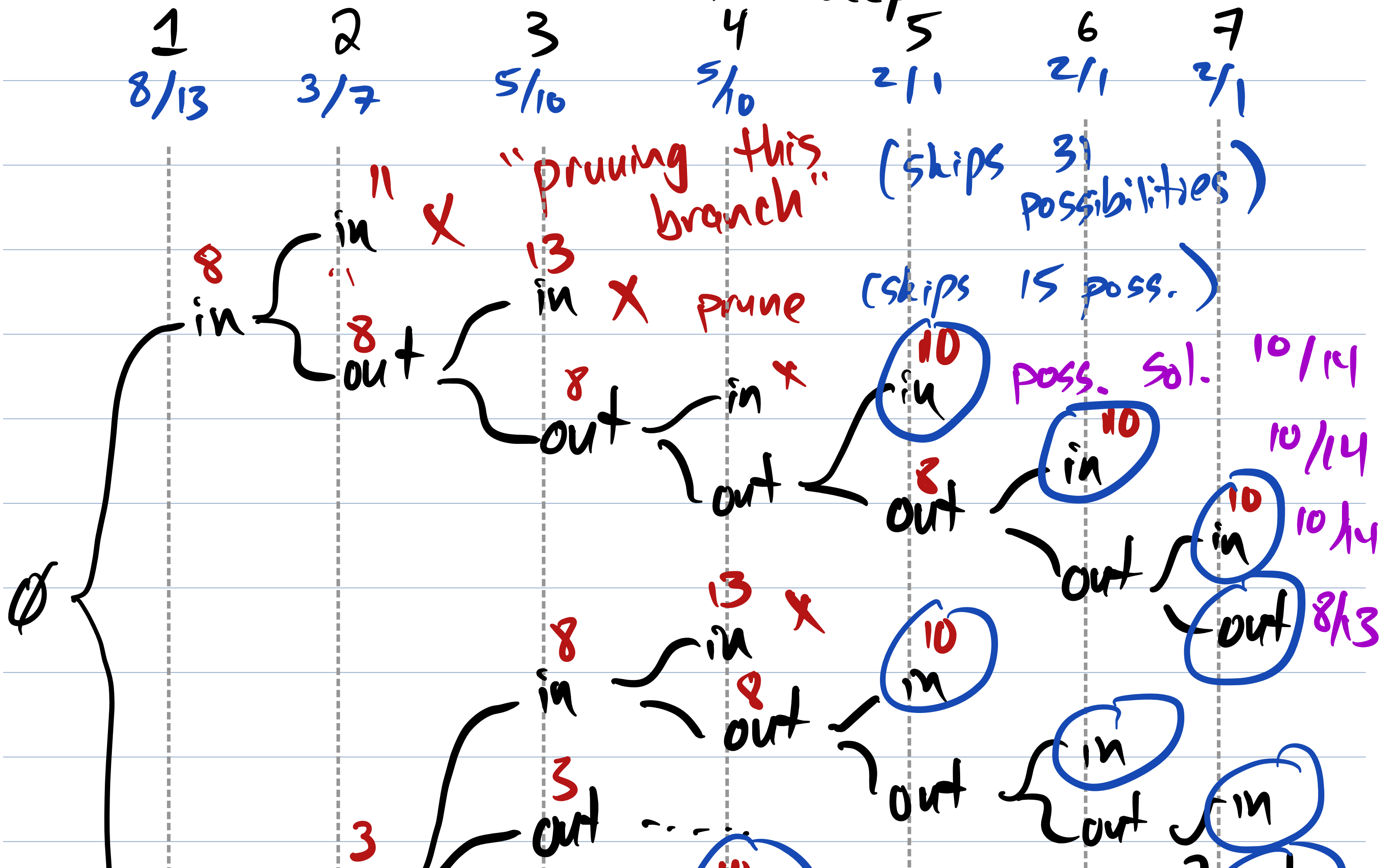
Monday, Feb 27, 2023
Lecture # 18
MSSC 6000

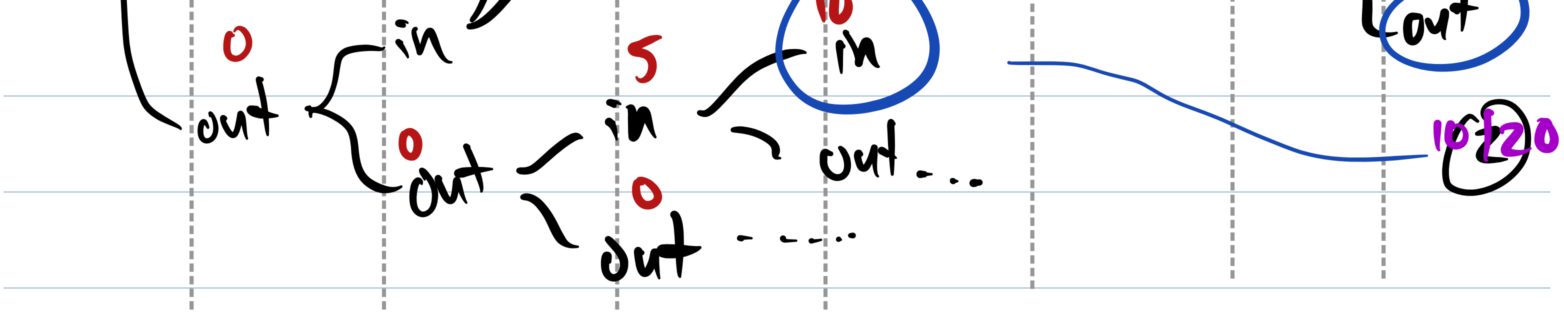
Announcements

Office hours today 1-2 in CU 307

- * HW 3 due Wed, March 8, 11:59pm
- * Office Hours on Wednesday are cancelled
- * Midterm Exam, Wed, March 8 in class

Backtracking: for each item, decide whether or not to keep it C=10





In code, recursion.

We'll have a function called
"solve"

input: * a list of remaining items } items_left
(weight, value) pairs
* capacity_left

output: the best solution as a list
of items (weight, value) pairs

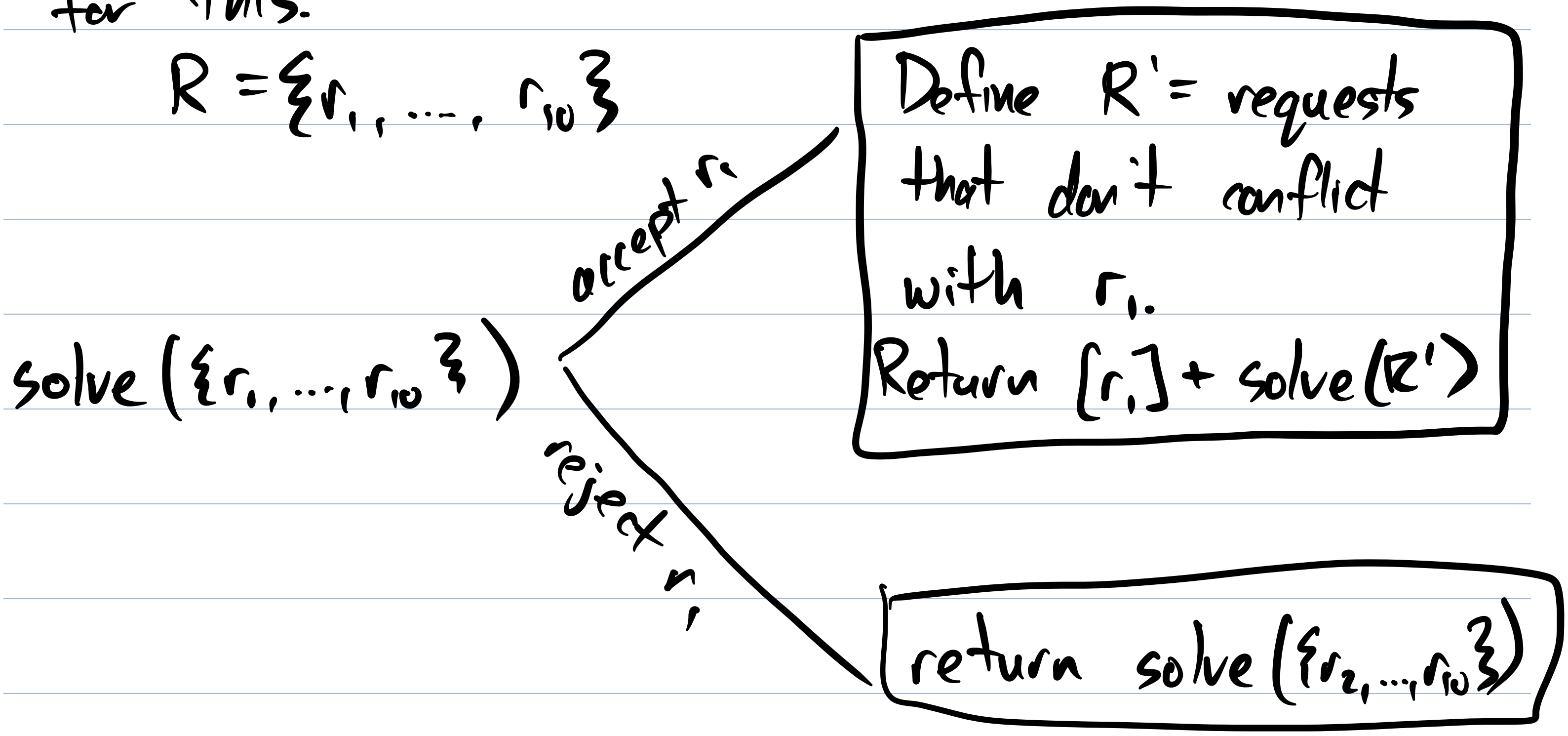
Ex 3: Weighted Interval Scheduling

Requests $R = \{r_1, r_2 r_3, \dots\}$

You either accept or reject each request.
If you accept r_i then in the future
you can ignore all requests that conflict

with r_i. Backtracking is perfect for this.

$R = \{r_1, \ldots, r_{10}\}$



Define R' = requests that don't conflict with r_1.
Return $[r_1]$ + solve(R')

return solve($\{r_2, \ldots, r_{10}\}$)

Pseudocode

```
function solve(requests):
    # output: best sol. made from [requests]
    if len(requests) = 0:
        return []
    new_request = requests[0]
    compatible = requests compatible with new_request
    accept_solution = [new_request] + solve(compatible)
    reject_solution = solve(requests[1:])
    return whichever of accept or reject is best
```