Homework 3 solutions are posted on Angel as pdf.
For Q2, you can use the Review mode in Angel.

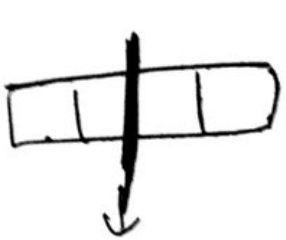
Next lecture is Midterm Review.
Midterm covers everything up to but not including Dynamic Programming.

$n \log n$ $\quad$ $n^{1.5}$ $\quad$ $\log n = o(n^d) \ \forall d > 0$

$n^{1.5}$ faster

Sample Midterm is posted and solutions and recitation on Monday will cover the solutions.

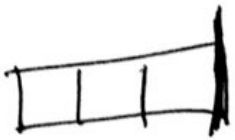
i	1	2	3	4
Pi	1	5	8	9.



p	1	5	8	10
s	1	2	3	2

2-2

n-2



n=4

s 2, 2

Weighted interval scheduling problem
(not in the textbook).

- Job j starts at s_j, finished at f_j, has value v_j

"compatible."

- Goal: find a max weight subset of mutually ~~exclusive~~ compatible jobs.

e.g. a, g or b and f but not b and c.

- Lets label jobs by finishing time:

$$f_1 \leq f_2 \leq \cdots \leq f_n.$$

- Brute Force: Check every ~~sub~~ schedule (2^n of them)
- What would be a natural subproblem to consider?

OPT(j) = value of the optimal solution to the problem consisting only of jobs 1, ..., j

Lets assume that by "magic" you knew the values for OPT(1) .... OPT(n-1). Then can you give a formula for OPT(n)?
Usually, based on some inclusion/exclusion argument or a "best choice" argument.

For computing OPT(j), observe that job j can either be or not be in the solution.

Case be - Find all jobs incompatible with job j. Let x be the last finish time of the remaining jobs. And let y be the largest index of remaining jobs.

Case not be - Then OPT(n) = OPT(n-1)

→ Then OPT(n) = V_j + OPT(y)