Good news!

Today	Tonight	Thursday	Thursday Night	Friday	Friday Night	Saturday	Saturday Night	Sunday
20%								
Slight Chc Showers	Decreasing Clouds	Sunny	Patchy Fog	Patchy Fog	Patchy Fog	Patchy Fog	Partly Cloudy	Mostly Sunny
Hi 50 °F	Lo 37 °F	Hi 70 °F	Lo 40 °F	Hi 75 °F	Lo 45 °F	Hi 81 °F	Lo 49 °F	Hi 73 °F

from forecast.weather.gov

Tech Rendezvous with Alumnae

Join us for an evening with alumnae, brought to you through the wonders of technology!

Grab a dinner on us and enjoy 5 short talks from a **panel of alumnae and friends** sharing a diverse set of career perspectives. Then check out the **student project showcase** as you mosey on over to our very own coffeehouse, specially created for this event! At the **virtual coffeehouse**, enjoy light refreshments and chat with alumnae about careers, life after graduation or ask for some advice on your own career path.

Program schedule

Dinner available starting at 5:30pm in Cleveland L2

6:00 - 7:00	**Virtual panel**	Cleveland L2
7:00 - 7:20	**Project showcase**	Kendade 307
7:20 - 8:20	**Virtual coffeehouse**	Kendade 307

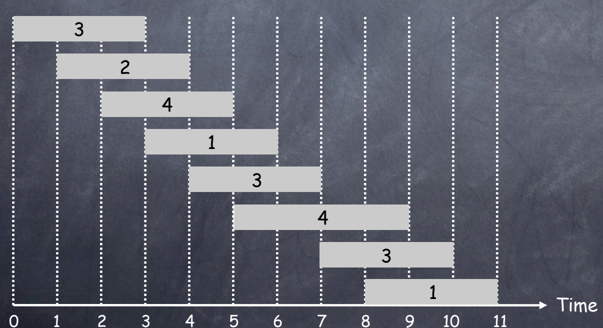
Dynamic Programming Formula

- Divide a problem into a polynomial number of smaller subproblems
- Solve subproblem, recording its answer
- Build up answer to bigger problem by using stored answers of smaller problems

3

Weighted Interval Scheduling

- Job j starts at s_j, finishes at f_j, and has weight or value v_j .
- Two jobs compatible if they don't overlap.
- Goal: find maximum weight subset of mutually compatible jobs.



4

Dynamic Programming Approach to Segmented Least Squares

- What are the subproblems?

  Best solution considering the jobs from 1 to j

- What is the recurrence we should use?

  - Which subproblems do we refer to in our recurrence?

Best solution from job 1 to j-1 AND best solution from job 1 to the last preceding job that does not conflict with job j

  - How do we choose the best solution of the current problem based on subproblem solutions?

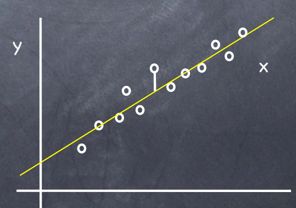
$$OPT(j) = \begin{cases} 0 & \text{if } j = 0 \\ \max\{\, \boxed{v_j + OPT(p(j))},\ \boxed{OPT(j-1)} \,\} & \text{otherwise} \end{cases}$$

5

Least Squares

- Foundational problem in statistic and numerical analysis.
  - Given n points in the plane: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$.
  - Find a line y = ax + b that minimizes the sum of the squared error:

$$SSE = \sum_{i=1}^{n} (y_i - ax_i - b)^2$$



6

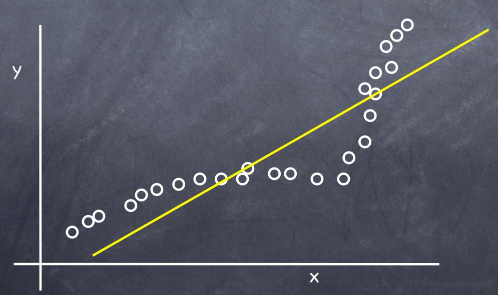
Least Squares Solution

- Result from calculus, least squares achieved when:

$$a = \frac{n \sum_i x_i y_i - (\sum_i x_i)(\sum_i y_i)}{n \sum_i x_i^2 - (\sum_i x_i)^2}, \quad b = \frac{\sum_i y_i - a \sum_i x_i}{n}$$

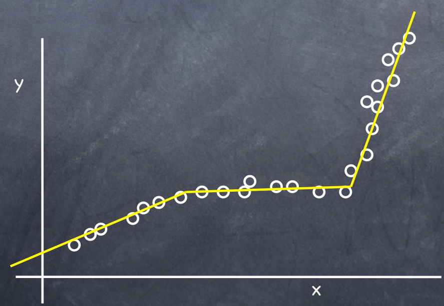
Least Squares

- Sometimes a single line does not work very well.



Segmented Least Squares

- Points lie roughly on a sequence of several line segments.
- Given n points in the plane $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ with $x_1 < x_2 < \ldots < x_n$, find a sequence of lines that fits well.



Dynamic Programming Approach to Segmented Least Squares

- What are the subproblems?

Best solution for a subset of the points from 1 to j

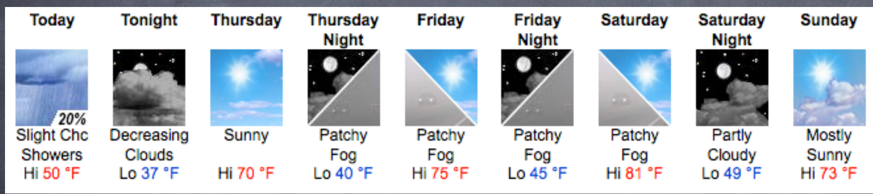
- What is the recurrence we should use?
  - Which subproblems do we refer to in our recurrence?

All possible line segments that end at the preceding point

  - How do we choose the best solution of the current problem based on subproblem solutions?

$$OPT(j) = \begin{cases} 0 & \text{if } j = 0 \\ \min\limits_{1 \le i \le j} \{\, e(i,j) + c + OPT(i-1) \,\} & \text{otherwise} \end{cases}$$

Good news!



Today	Tonight	Thursday	Thursday Night	Friday	Friday Night	Saturday	Saturday Night	Sunday
20%								
Slight Chc Showers	Decreasing Clouds	Sunny	Patchy Fog	Patchy Fog	Patchy Fog	Patchy Fog	Partly Cloudy	Mostly Sunny
Hi 50 °F	Lo 37 °F	Hi 70 °F	Lo 40 °F	Hi 75 °F	Lo 45 °F	Hi 81 °F	Lo 49 °F	Hi 73 °F

from forecast.weather.gov

11

The Price is Right!

(or shopping with somebody else's money)

- Spend as much money as possible without going over $100. You can buy at most 1 of each.

CD	$18	Jeans	$40
DVD	$35	Dinner	$15
Book	$8	Ice cream	$5
Shoes	$61	Pizza	$7

12

Dynamic Programming Approach

- What are the subproblems?
- What is the recurrence we should use?
  - Which subproblems do we refer to in our recurrence?
  - How do we choose the best solution of the current problem based on subproblem solutions?