COL702: Advanced Data Structures and Algorithms

Thanks to Miles Jones, Russell Impagliazzo, and Sanjoy Dasgupta at UCSD for these slides.

OPTIMIZATION PROBLEMS

In general, when you try to solve a problem, you are trying to find the best solution from among a large space of possibilities.

Format for an optimization problem:

- Instance: what does the input look like?
- Solution format: what does an output look like?
- Constraints: what properties must a solution have?
- Objective function: what makes a solution better or worse?

EXAMPLE

SHORTEST PATH

- Instance
- Solution format
- Constraint
- Objective

EXAMPLE

SHORTEST PATH

- Instance: Graph G with positive edge lengths l(e); vertices s,t
- Solution format: list of edges $e_1,\ldots,e_k$
- Constraint: must form a path from s to t
- Objective: minimize $\sum l(e_j)$

THE SEARCH SPACE

In general, there will be **exponentially** many possible solutions.

- The number of paths in a graph from s to t
- The number of distinct orderings of the vertices
- The number of cycles in a graph
- The number of spanning trees of a graph

GLOBAL SEARCH VS LOCAL SEARCHES

- In general, exponentially many possible solutions.

- Obvious algorithm: try them all and take the best.
  This is usually prohibitively slow
  Sometimes unavoidable (unless P=NP)

- For efficiency: ***break the massive global search for a solution into a series of simpler local searches for parts of the solution***.
  Which edge do we take first? Then second? …

- If you can't tell which local choice is best, may still have to use **exhaustive search** to try out all combinations of local decisions and find the optimal one.

THE GREEDY METHOD

- In *some* cases (not all!!!), there is sufficient structure that allows you to reach the correct solution by just picking the straightforward “best” decision at each stage.

- This is called the **Greedy Method**.

- It doesn’t always work.

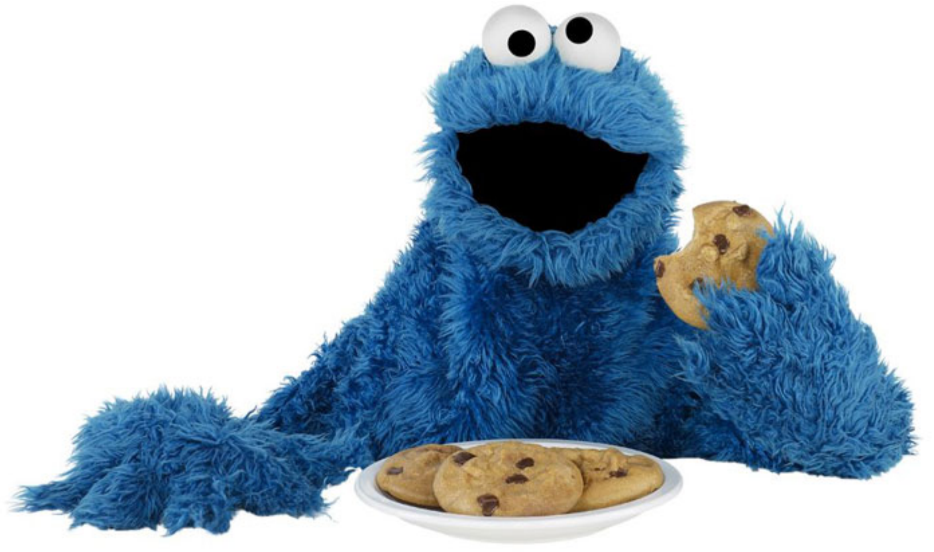
- Just as in life, acting in one’s immediate best interest is not always the best longer-term strategy.

OTHER USES OF LOCAL DECISIONS

Many of the other techniques we'll study are also based on breaking up global search into local decisions:

- Backtracking
- Dynamic programming
- Hill-climbing
- Stochastic search heuristics

COOKIES

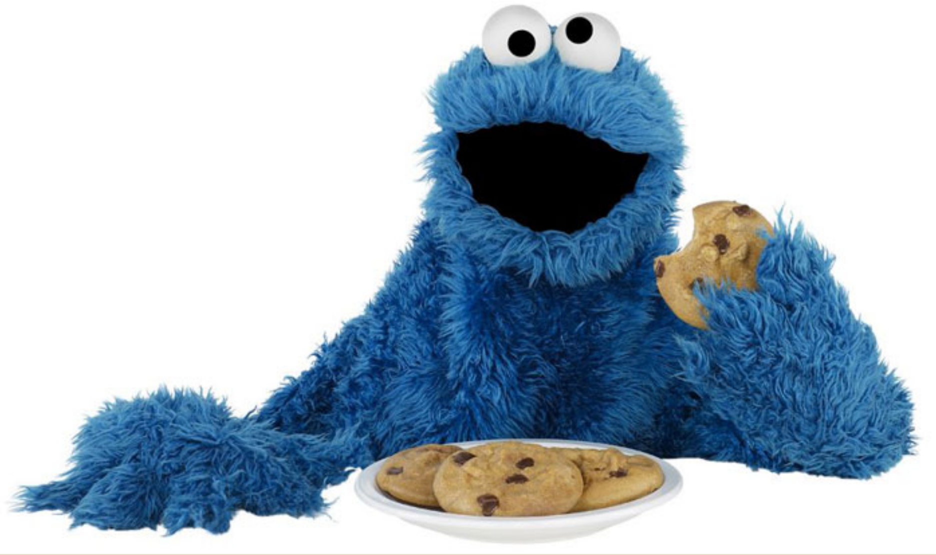


- You are the cookie monster and you have a 6x6 tray of freshly baked cookies in front of you. They are all chocolate chip but may have different sizes.
- If you are only allowed to take six cookies, how can you maximize your total cookie intake?
- Devise an algorithm to do this.

COOKIE PROBLEM SPECIFICATION

- Instance:
- Solution format:
- Constraints:
- Objective:

COOKIES

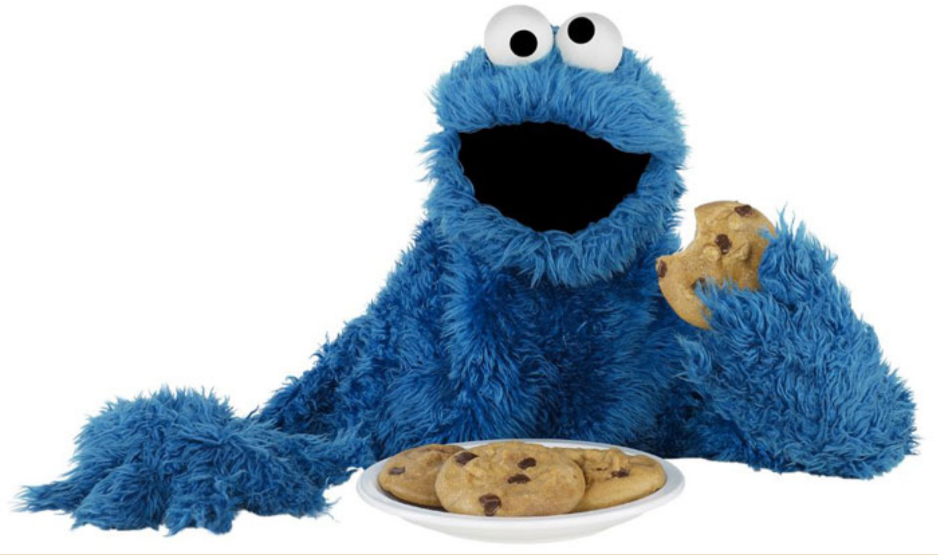


56	76	69	60	75	51
61	77	74	72	80	58
82	97	94	88	99	92
47	68	59	52	65	40
78	81	79	71	85	62
50	67	73	57	70	46

1. What is an algorithm you could use to select the *best* option?

(The best option means that the sum of all the cookie's sizes is the highest possible.)

COOKIES



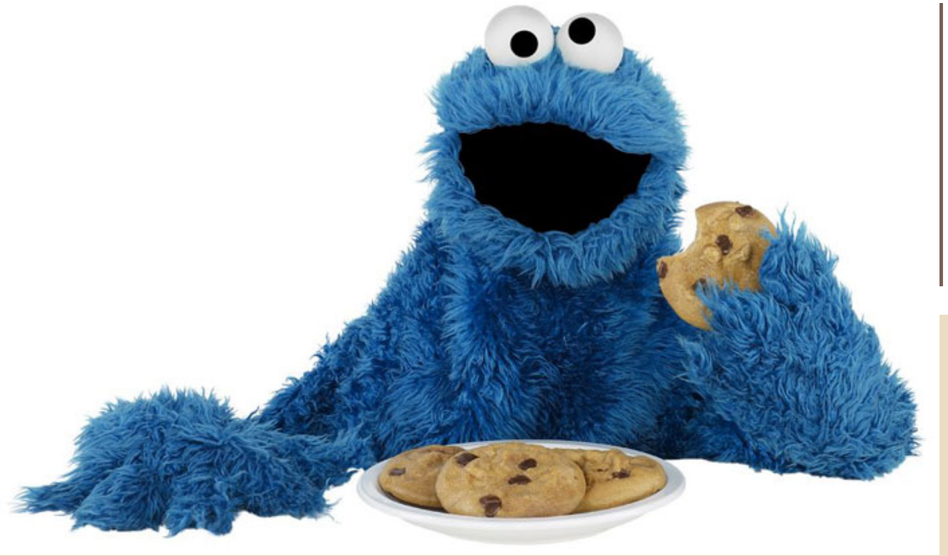
56	76	69	60	75	51
61	77	74	72	80	58
82	**97**	**94**	**88**	**99**	**92**
47	68	59	52	65	40
78	81	79	71	**85**	62
50	67	73	57	70	46

1. What is an algorithm you could use to select the *best* option?

(The best option means that the sum of all the cookie's sizes is the highest possible.)

99+97+94+92+88+85=555

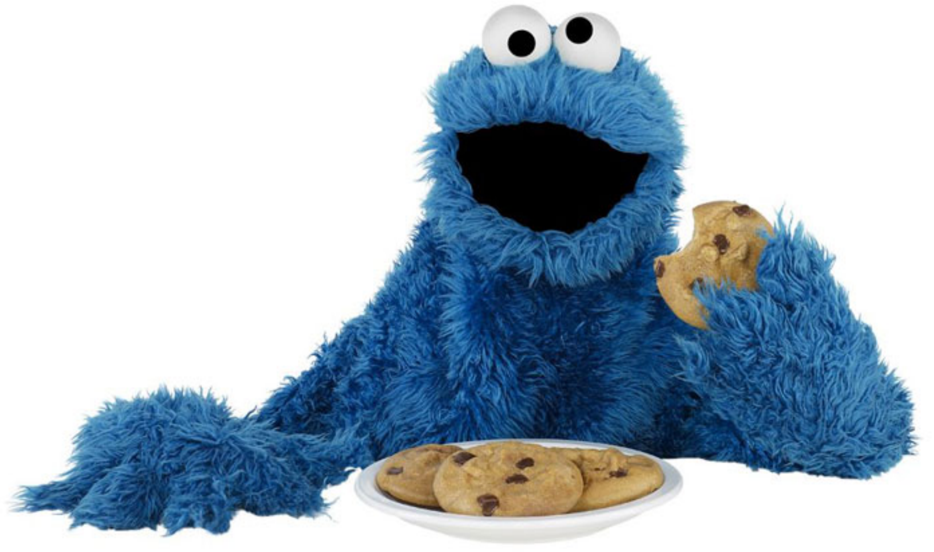
COOKIES



56	76	69	60	75	51
61	77	74	72	80	58
82	97	94	88	99	92
47	68	59	52	65	40
78	81	79	71	85	62
50	67	73	57	70	46

2. What is an algorithm you could use to select the *best* option if you can only select one cookie from each row?

COOKIES



56	**76**	69	60	75	51
61	77	74	72	**80**	58
82	97	94	88	**99**	92
47	**68**	59	52	65	40
78	81	79	71	**85**	62
50	67	**73**	57	70	46

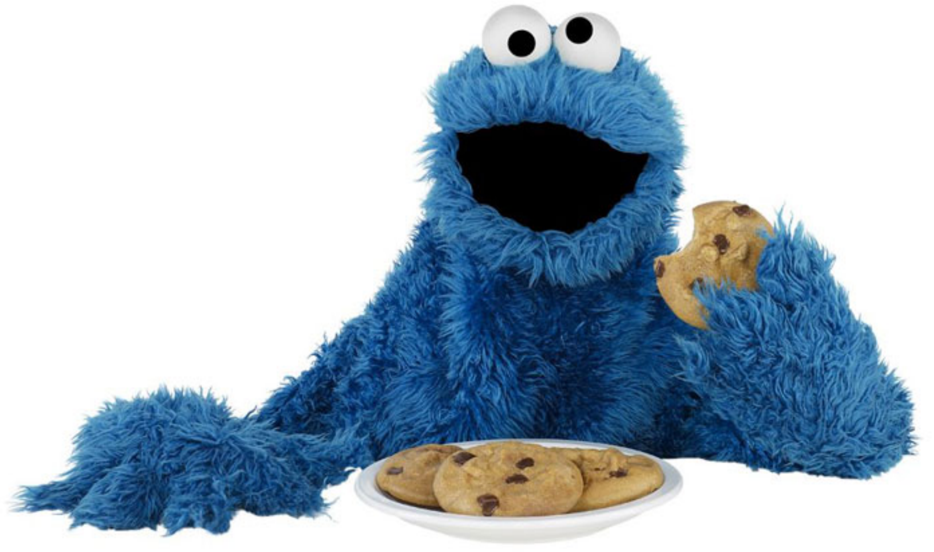
2. What is an algorithm you could use to select the *best* option if you can only select one cookie from each row?

76+80+99+68+85+73=481

ONE PER ROW COOKIE PROBLEM SPECIFICATION

- Instance:
- Solution format:
- Constraints:
- Objective:

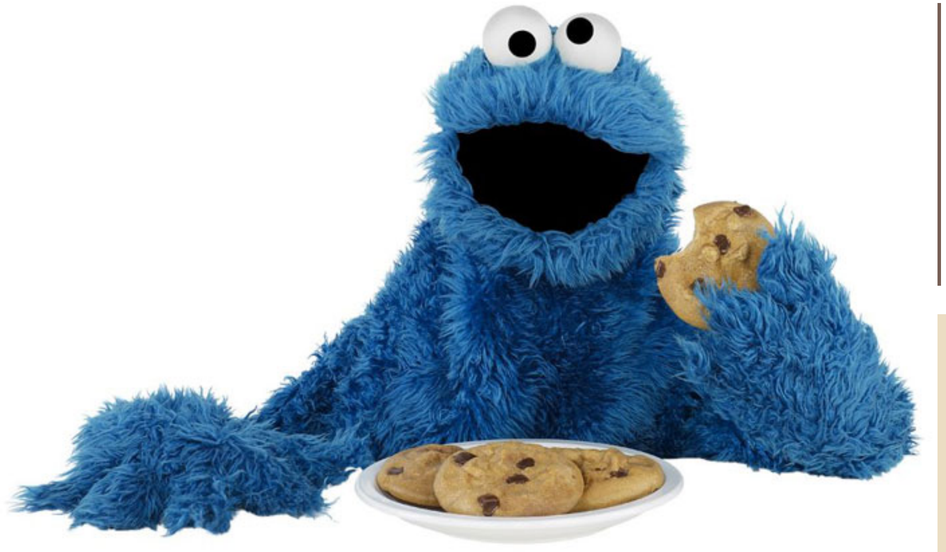
COOKIES



56	76	69	60	75	51
61	77	74	72	80	58
82	97	94	88	99	92
47	68	59	52	65	40
78	81	79	71	85	62
50	67	73	57	70	46

3. What is an algorithm you could use to select the *best* option if you can't select 2 cookies from the same row or column?

COOKIES



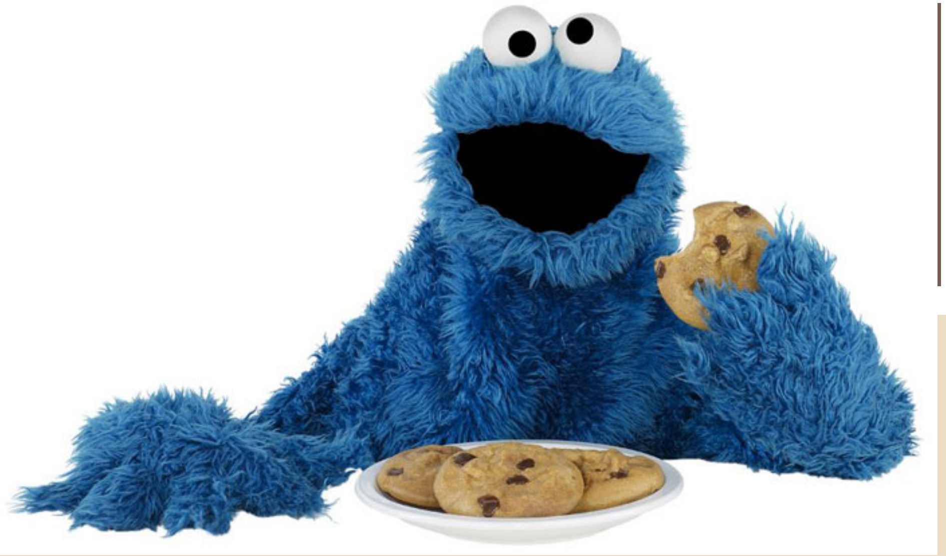
56	76	69	60	75	51
61	77	74	72	80	58
82	97	94	88	**99**	92
47	68	59	52	65	40
78	81	79	71	85	62
50	67	73	57	70	46

3. What is an algorithm you could use to select the *best* option if you can’t select 2 cookies from the same row or column?

ONE PER ROW & COLUMN COOKIE PROBLEM SPECIFICATION

- Instance:
- Solution format:
- Constraints:
- Objective:

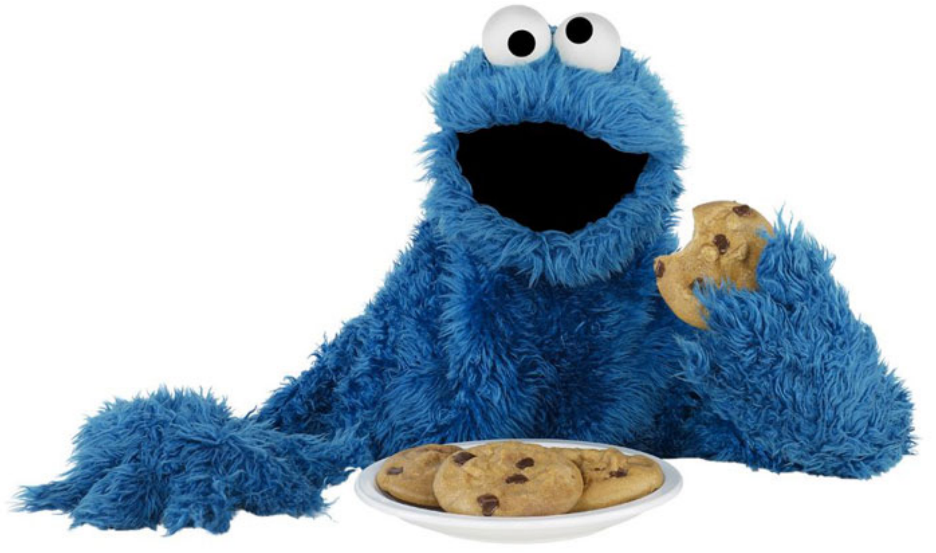
COOKIES



56	76	69	60	75	51
61	77	74	72	80	58
82	97	94	88	99	92
47	68	59	52	65	40
78	81	79	71	85	62
50	67	73	57	70	46

3. What is an algorithm you could use to select the *best* option if you can't select 2 cookies from the same row or column?

COOKIES

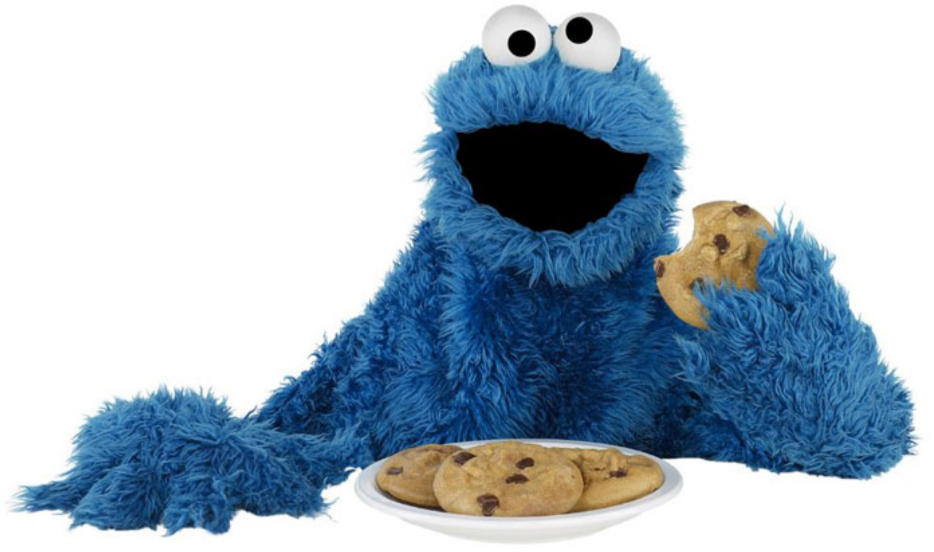


56	76	69	60	75	51
61	77	74	72	80	58
82	97	94	88	99	92
47	68	59	52	65	40
78	81	79	71	85	62
50	67	73	57	70	46

3. What is an algorithm you could use to select the *best* option if you can't select 2 cookies from the same row or column?

99+81+74+60+50+40=404

COOKIES



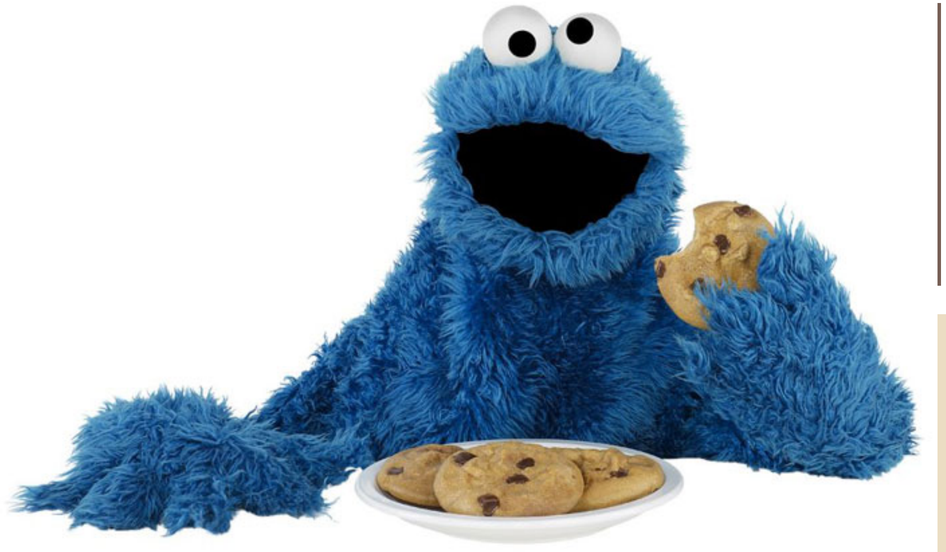
56	76	69	60	75	51
61	77	74	72	80	58
82	97	94	88	99	92
47	68	59	52	65	40
78	81	79	71	85	62
50	67	73	57	70	46

3. What is an algorithm you could use to select the *best* option if you can't select 2 cookies from the same row or column?

99+81+74+60+50+40=404

99+81+72+69+47+46=414

COOKIES



56	76	69	60	75	51
61	77	74	72	80	58
82	97	94	88	99	92
47	68	59	52	65	40
78	81	79	71	85	62
50	67	73	57	70	46

3. What is an algorithm you could use to select the *best* option if you can't select 2 cookies from the same row or column?

99+81+74+60+50+40=404

99+81+72+69+47+46=414

92+78+75+73+72+68=458!!!

THE GREEDY METHOD

- The Greedy Method does not always work.
- Because of this, when using the Greedy Method, we must **prove** the correctness of the algorithm.
- Or else, we must present a **counterexample** to show that a particular greedy method will not work.

CHOOSING BETWEEN GREEDY STRATEGIES

- For a single problem, there may be more than one potential greedy strategy: more than one way to choose the “best” possible choice at each step.
- Some of these strategies might work while others don’t. To sort this out, we use proofs and counterexamples.

IMMEDIATE BENEFIT VS OPPORTUNITY COSTS

IMMEDIATE BENEFIT:
How much does the choice we're making now contribute to the objective function?

OPPORTUNITY COST:
How much does the choice we're making now restrict future choices?

Usual Greedy strategy: Take best immediate benefit and ignore opportunity costs.

Greedy is optimal: Best immediate benefits outweigh opportunity costs.

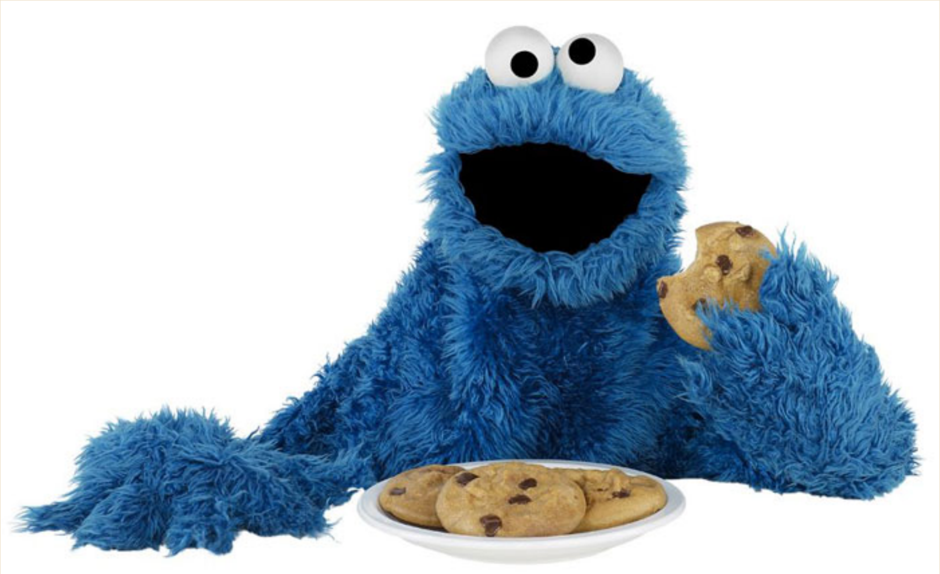
ONE PER ROW COOKIES

56	76	69	60	75	51
61	77	74	72	80	58
82	97	94	88	99	92
47	68	59	52	65	40
78	81	79	71	85	62
50	67	73	57	70	46

99 — Immediate benefit

82, 97, 94, 88, 92: Opportunity costs

(Since we can have at most one of these:97)



Immediate benefit > Opportunity cost

ONE PER ROW& COLUMN COOKIES

56	76	69	60	75	51
61	77	74	72	80	58
82	97	94	88	99	92
47	68	59	52	65	40
78	81	79	71	85	62
50	67	73	57	70	46

75
80
65
85
70

99

Immediate benefit

82, 97, 94, 88, 92: Opportunity costs

(Lose out on one in row, one in column: 92+85)

Immediate benefit < Opportunity cost