Fri, Mar 1, 2024

Scientific Computing

Announcements:

- → HW 3 due Fri, March 8
- → Wed March 6: In-class midterm

→ Review my HW 2 feedback!

→ Backtracking as a concept is on the exam but you won't need to code it

→ itertools

└ powerset

return itertools. ____

return list( )

Lecture 7 - Backtracking (continued)

Ex #1: Knapsack

Capacity: 10

With brute force:

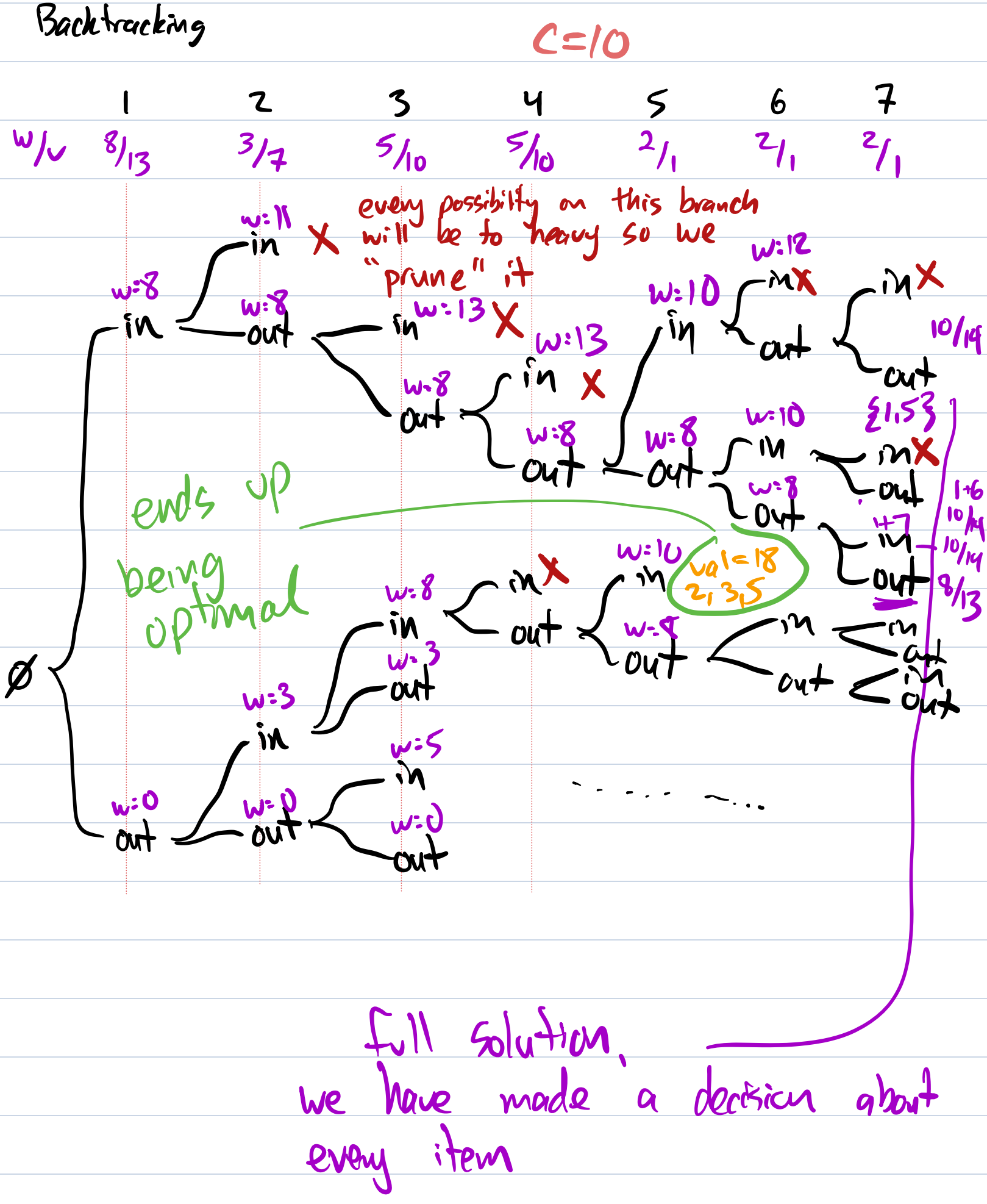
Possibilities: $\emptyset$, $\{1\}$, $\{2\}$, ...

$\{1,3,4,5,7\}$

too heavy, and still too heavy if you remove any single item

item	weight	value
1	8	13
2	3	7
3	5	10
4	5	10
5	2	1
6	2	1
7	2	1

Backtracking



What are we doing?

- Putting a hierarchy on decisions that builds the whole search space with the critical property: if a candidate (partially built solution) is bad, then every candidate that comes after it (full solutions built from that partial solution) must also be bad.

Backtracking checks or rules out every candidate in the search space.

In the worst case it's as slow (or even a little slower) than brute force, but in practice significantly faster.

3 items

	w	v
1	5	10
2	3	8
3	4	7

$C = 10000000$