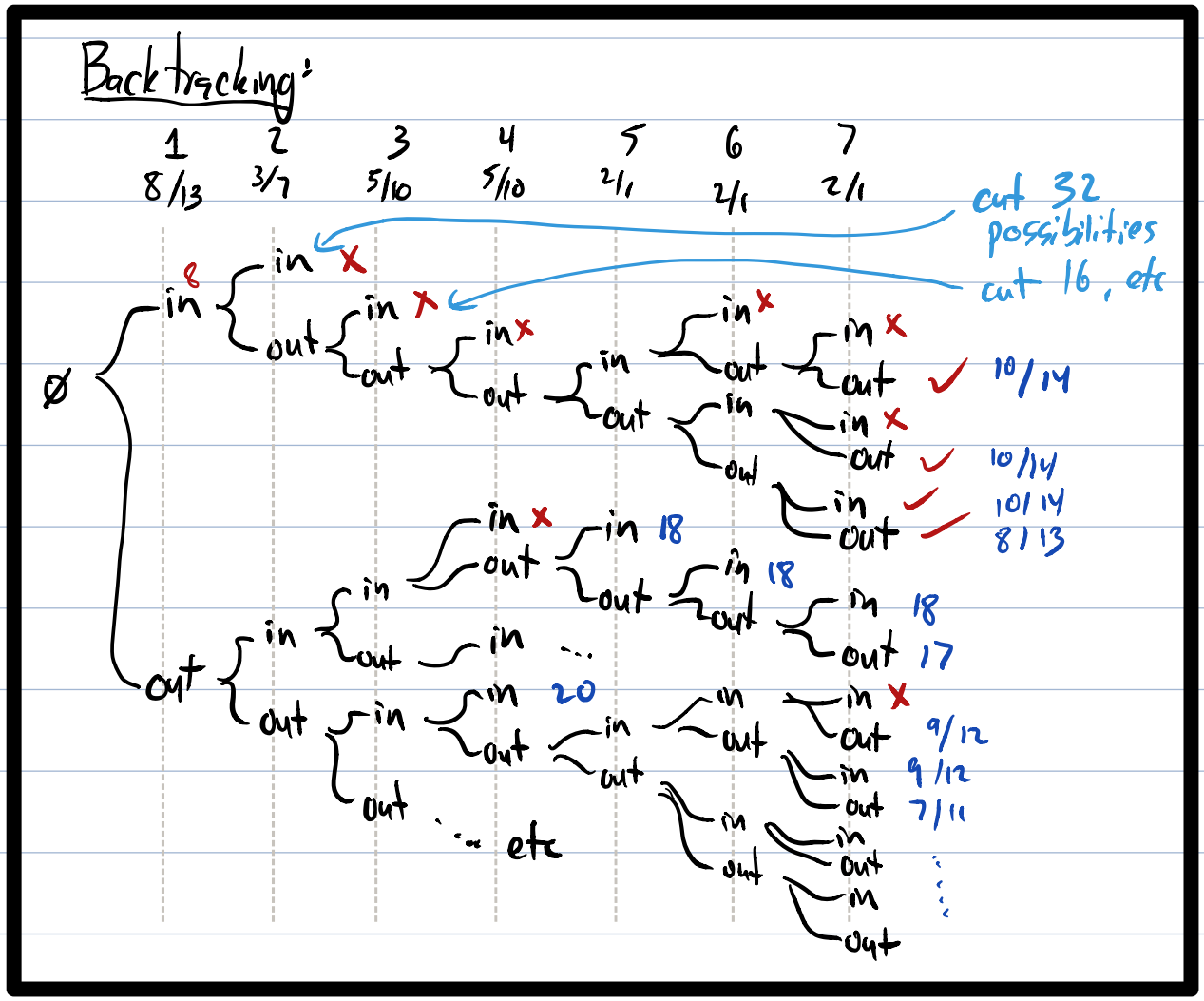

Backtracking
1 2 3 4 5 6 7
8/13 3/7 5/10 5/10 2/1 2/1 2/1
out 32 possibilities
cut 16, etc
Ø
in
out
8
in x
out
in x
out
in x
out
in
out
in x
out
in
out
in x
out
in x
out
in
out
in
out
in x
out
10/14
10/14
10/14
8/13
in x
out
in 18
out
in 18
out
in 18
out 17
in
out
in ...
in
out
in 20
out
in
out
in
out
in x
out
9/12
in 9/12
out 7/11
in
out
in
out
... etc

Fri, March 5 Lecture #18

Announcements

→ HW 2 graded
→ HW 3 posted (due Mon, March 15)

Relaxation

Knapsack

Capacity: 14

item	weight	value	
1	8	13	0.5
2	3	7	1
3	5	10	1
4	5	10	0.4
5	2	1	
6	2	1	
7	2	1	

Fractional Knapsack: You're allowed to take fractions of items

4 / 6.5
3 / 7
5 / 10
2 / 4
14 / 27.5

Theorem: A greedy and optimal solution can be found by:

(1) order the items by value density
(2) take items from the top in full until you can't anymore

(3) take whatever fraction you can of the next item

We won't prove this, but think about it until you believe it.

Capacity: 14

item	weight	value	density = value/weight		
1	8	13	1.625	1/8	1 / 13/8
2	3	7	2.333	1	3/7
3	5	10	2	1	5/10
4	5	10	2	1	5/10
5	2	1	0.5		14/
6	2	1	0.5		28.625
7	2	1	0.5		

If capacity = 10: value = 21 which is better than regular knapsack.

So, Fractional Greedy = Fractional Optimal ≥ Regular Optimal

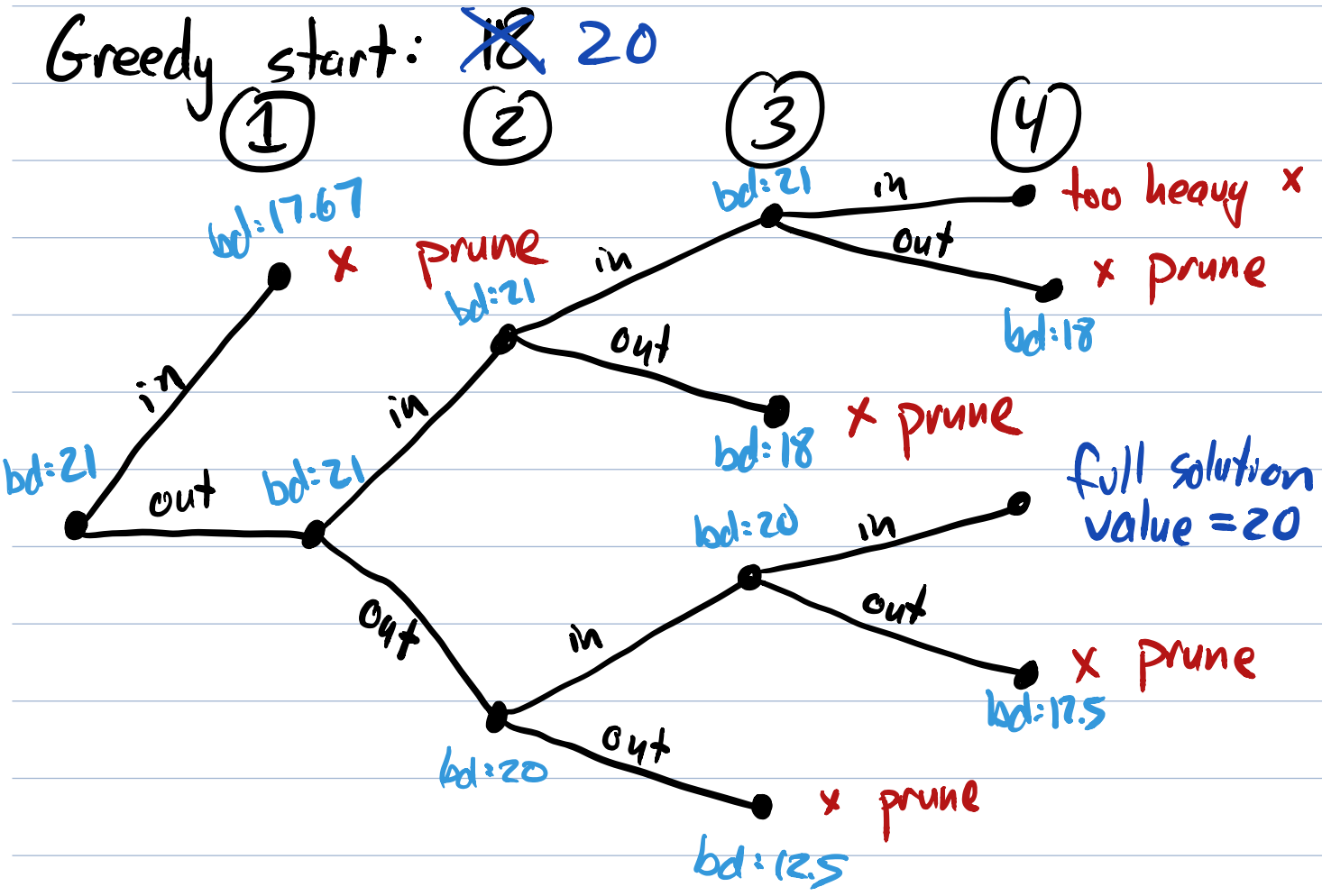
Thus, we can get an UB for B+B

by computing the greedy fractional solution on whatever items are left to decide.

Capacity: ~~11~~ 10

item	weight	value	density		weight of 7 left
~~1~~	~~8~~	~~13~~	~~1.625~~		
2	3	7	2.333	3/7	14
3	5	10	2	1 5/10	10
4	5	10	2	0.4 2/4	18
5	2	1	0.5	10/21	
6	2	1	0.5		
7	2	1	0.5	UB	

If item 1 is out and item 2 is in, then we don't know how good we can do from here, but we definitely can't beat 21.

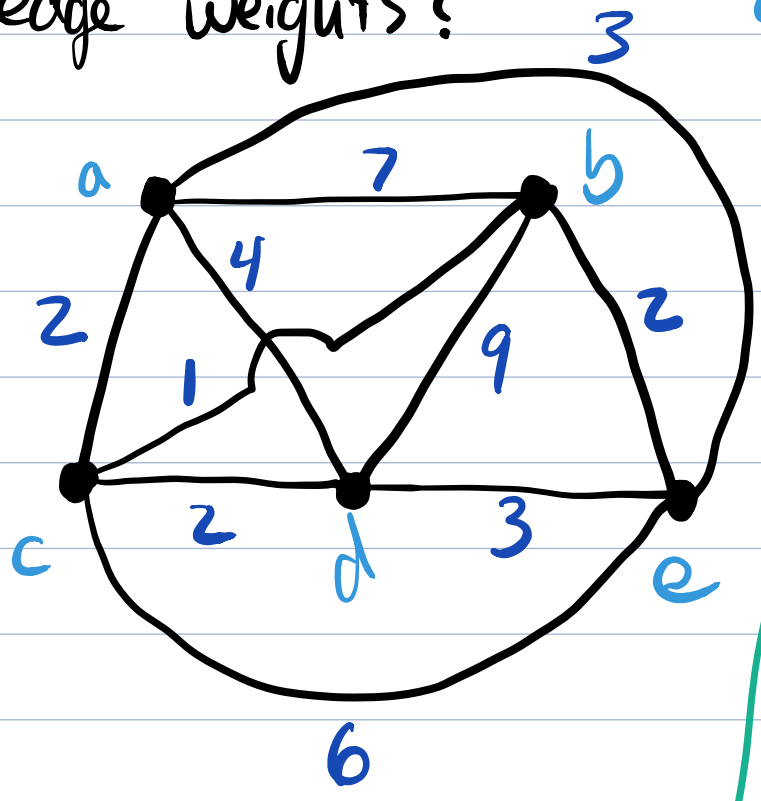


Problem #6: Travelling Salesman Problem (TSP)

There are n cities that a salesman needs to visit, and the return to the start. What is the shortest route ("tour") that visits each city exactly once and returns back to the starting place?

More formally: Consider a weighted graph G. Which ordering of the vertices gives you the smallest sum of the

edge weights?



$a \to d \to e \to c \to b \to a$

$4 + 3 + 6 + 1 + 7$

$= 21$

$a \to c \to b \to e \to d \to a$

$2 + 1 + 2 + 3 + 2$

$= 10$

$b \to e \to d \to a \to c \to b$

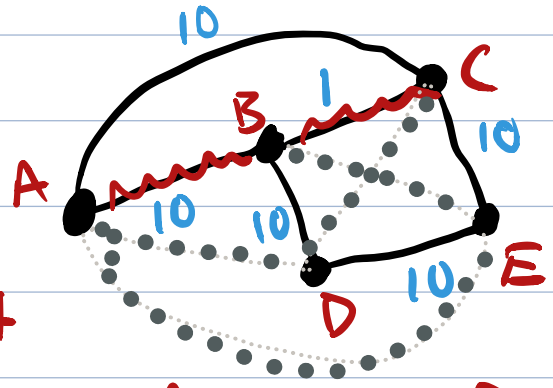
Same tour $= 10$

Greedy algo's
* pick any start vertex v_1
* pick v_2 to be the vertex that is closest to v_1
* pick v_3 to be the closest unused vertex to v_2
* repeat until out of vertices, then go back to v_1.

Only guaranteed to work if all edges

are present.

Ex:



A → B → C → E → D → stuck

A → C → B → stuck

fails for any start

- Brute force:
  - start city doesn't matter
  - second city: n-1
  - third city: n-2
  - ⋮
  - last city: 1

Search space: $(n-1)!$ (n^n)

$O((n-1)!)$

Dynamic Programming:

$O(n^2 \cdot 2^n)$

Monday: B + B.