Fri, March 11 - Day 21

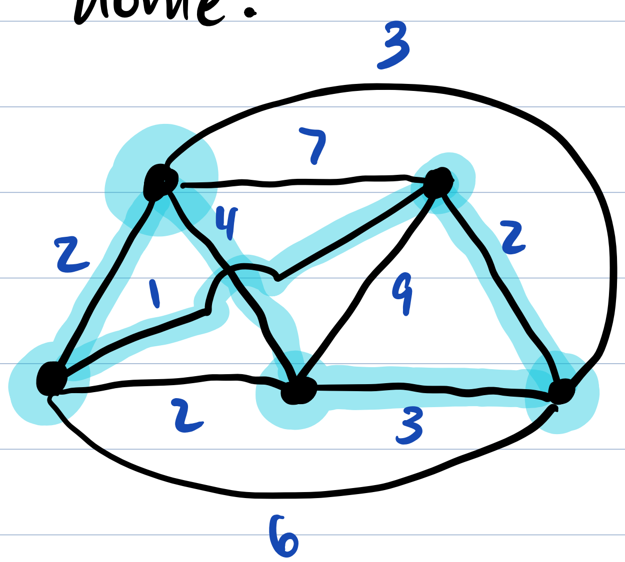
→ Midterm Exam, Wed after break
→ Takehome portion due following Monday
→ Covers up to and including backtracking (no B+B)

Topic 8 - Branch and Bound (continued)

→ Last class: Knapsack ⇝ relaxation

→ Today: Traveling Salesman Problem

n cities that a salesman needs to visit, and then return home.
What is the shortest route to visit each city exactly once and return home?
"tour"



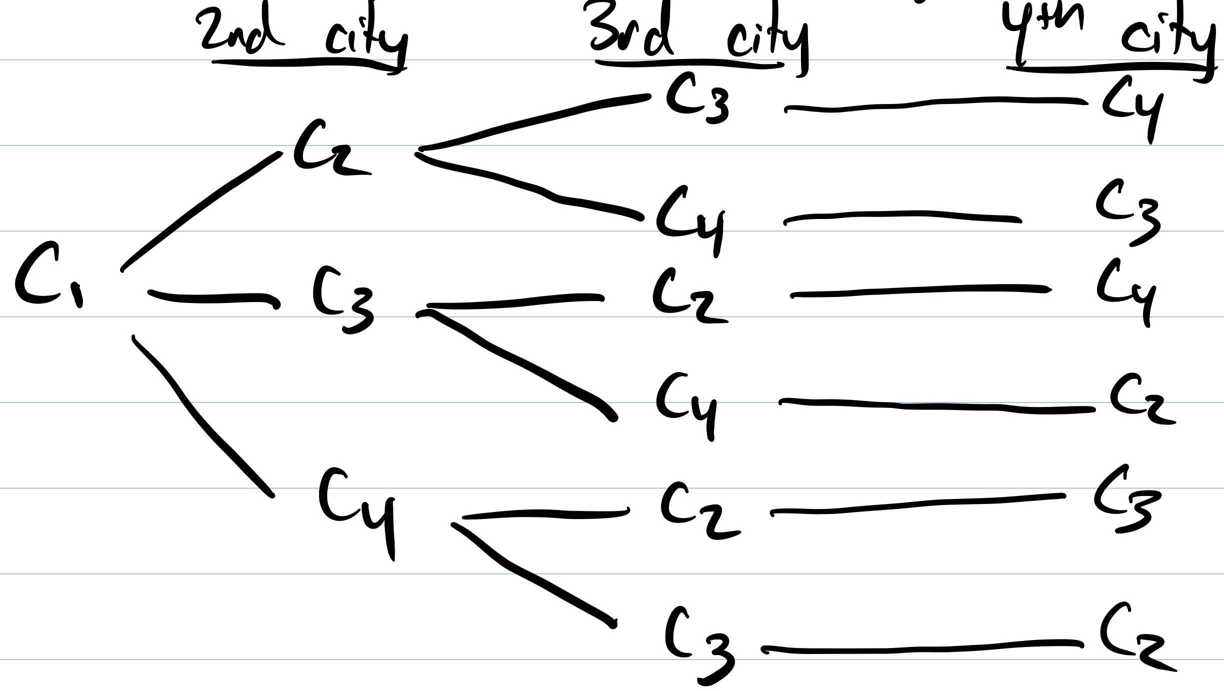
Minimize cost.

$2+1+2+3+4 = 12$

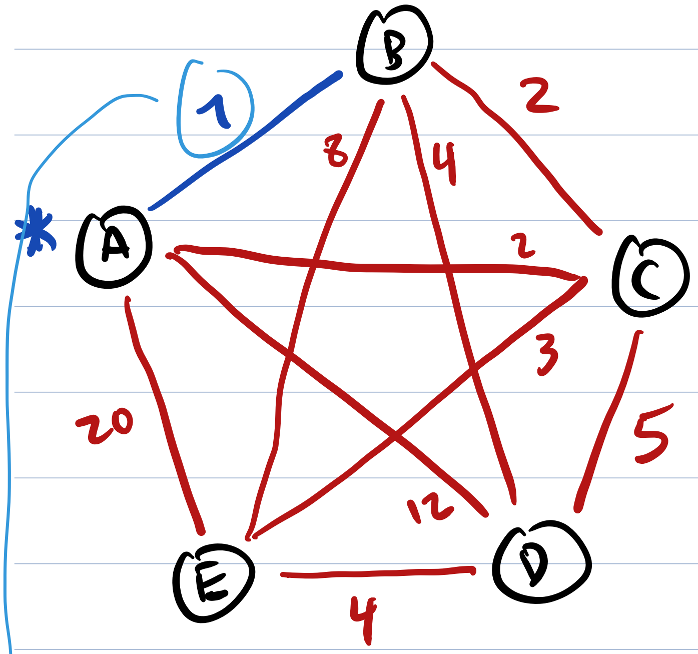
We are <u>minimizing</u>, so greedy solutions are an upper bound on the optimal solution.

For branch-and-bound we want a lower bound.

Branch: assume 4 cities. $\{C_1, C_2, C_3, C_4\}$
Call C_1 the start city.



<u>Lower Bound:</u>
Suppose we have decided the first few steps of a tour. I don't know cheaply I can finish this tour, but <u>know for sure</u> that I can't do it cheaper than X.



A → B

We're definitely going to leave B. Cheapest = \$2

We definitely have to enter and exit C at some point: \$4

D: \$8

E: \$7

Re-enter A: \$2

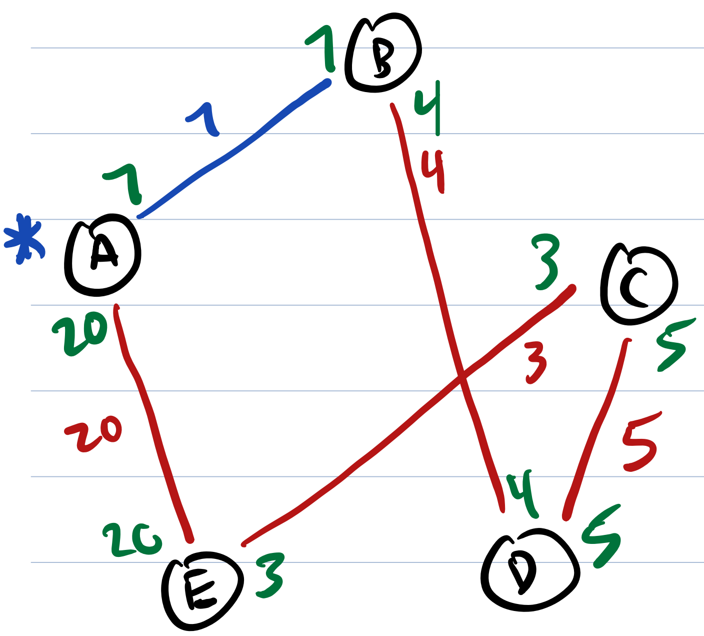
~~Wrong~~

Lower bound:

$7+2+4+8+7+2$

$=30$

Double counts the costs. When you exit B, you enter some other node.



Let T be some given tour.

If you add the cost going into or out of each city, you get double the total cost.

$$\text{cost}(T) = \frac{1}{2} \cdot \sum_{\substack{v \in V \\ \text{V = set of all vertices}}} \left([\text{cost to enter } v] + [\text{cost to exit } v] \right)$$

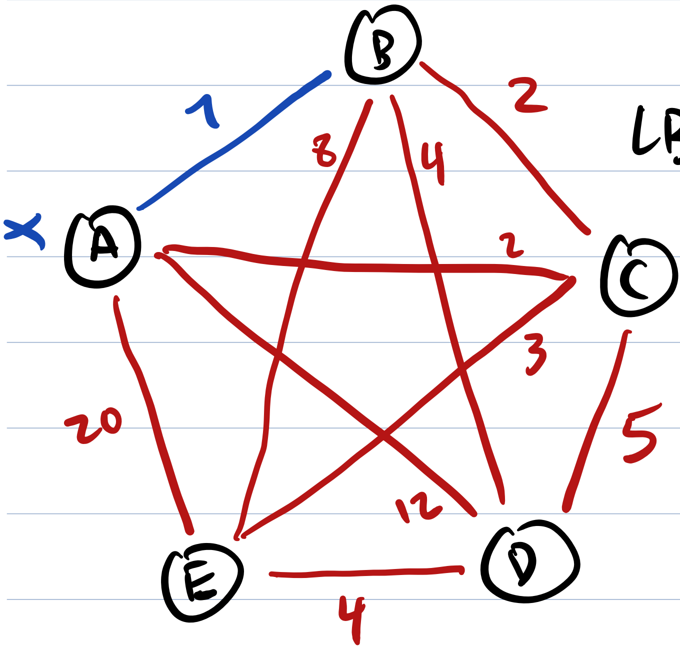
Now suppose we're considering some subspace S (we've already picked the first few cities) and we want a lower bound on the cost of all tours in S.

Pick $T \in S$ arbitrarily.

$$\text{cost}(T) = \frac{1}{2}\Big(\underbrace{[\text{enter } v_1] + [\text{exit } v_1]}_{\geq \text{ sum of two cheapest edges attached to } v_1} + \underbrace{[\text{enter } v_2] + [\text{exit } v_2]}_{\geq \text{ sum of two cheapest edges attached to } v_2}$$

$$+ \cdots + \underbrace{[\text{enter } v_n] + [\text{exit } v_n]}_{\geq \text{ sum of 2 cheapest edges attached to } v_n} \Big).$$

$\geq \frac{1}{2}$(sum of: for each vertex, use any edges you've already decided on, plus cheapest remaining, to get to two total)



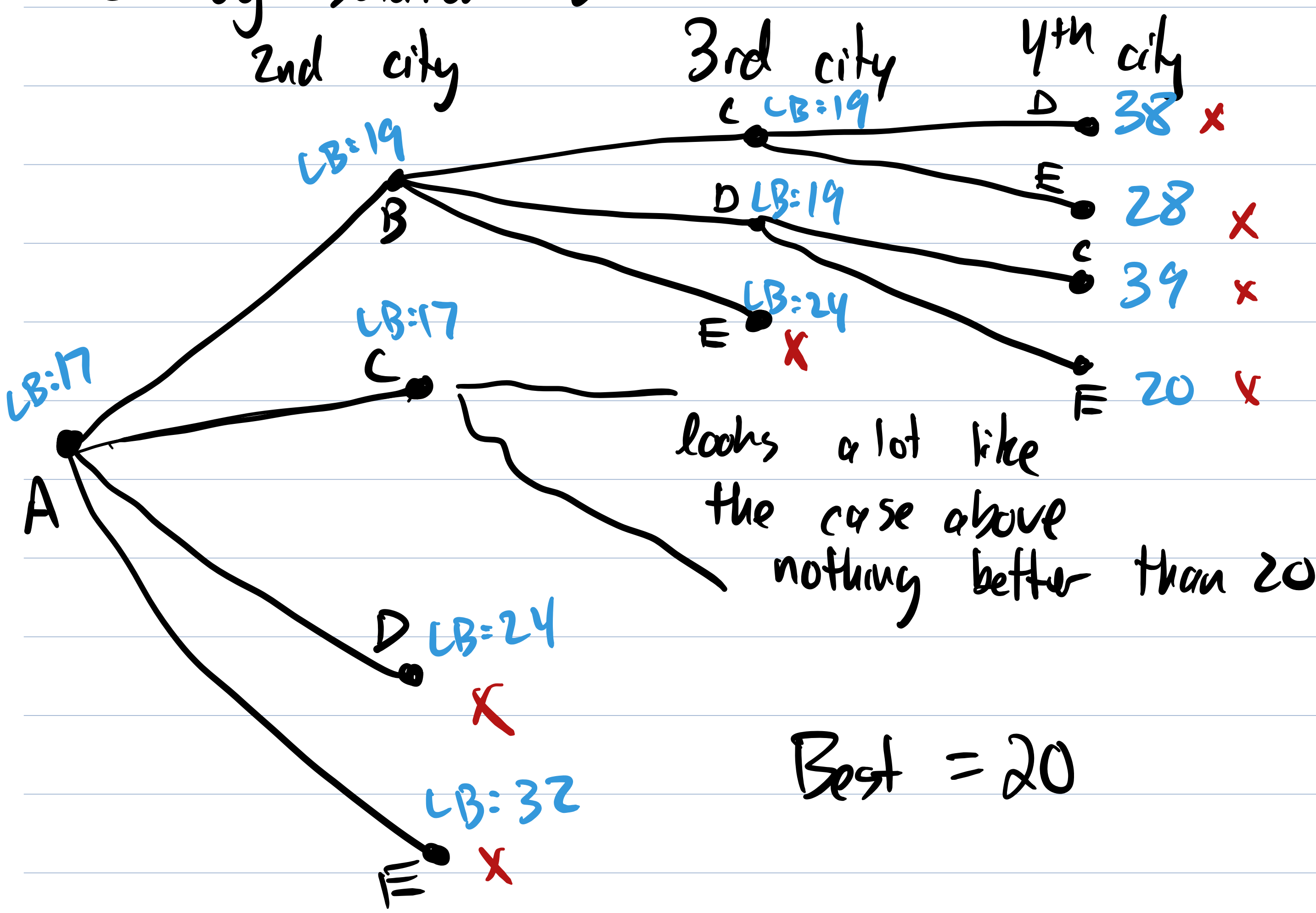
already chosen

$$LB = \frac{1}{2}\left(\underbrace{7^{*}+2}_{A} + \underbrace{7^{*}+2}_{B} + \underbrace{2+2}_{C} + \underbrace{4+4}_{D} + \underbrace{4+3}_{E}\right)$$

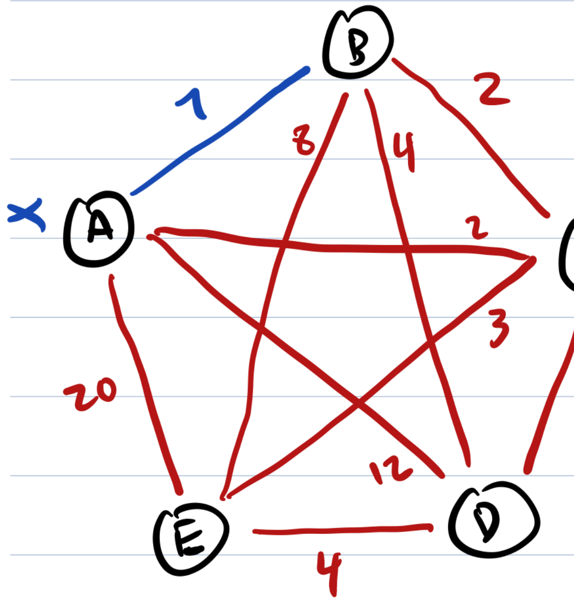
$$= \frac{1}{2}(37) = 18.5$$

Edges all integer weights, so if 18.5 is a lower bound, then so is 19.

Greedy solution: 20



Best = 20



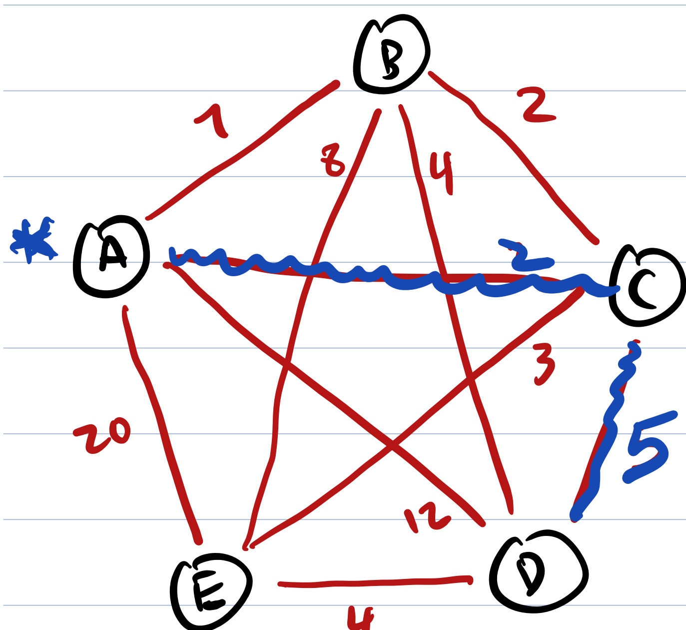
LB =

Would picking a different start node have helped?

You'll get the same answer, but possibly with more or less pruning.

Picking A first means we don't think about re-entering A until the very end.

One more small improvement to the bound.



not possible b/c C is full

A: 2, 7
B: ~~2~~, 4, 7
C: 2, 5
D: 4, 5
E: ~~3~~, 4, 8

increases bound by 5

[illegible]