Asymptotic Notation

Merge sort is $O(n^3)$

TRUE FALSE

$f(n)$ is $O(g(n))$

$\exists c, n_0$ s.t. $\forall n \geq n_0$, $f(n) \leq c \cdot g(n)$

Algorithm A has worst case $\Theta(n^2)$
B has worst case $\Theta(n \log n)$

	T	F
B is necessarily faster than A on all inputs		✓
B is faster than A on all sufficiently large inputs	✓	
For sufficiently large n, there is always an input of size n for which B is faster than A	✓	

Data Structures

Abstract Data Type

LIST

implementations

operations	array	doubly-linked linked list
add to front	O(n)	O(1)
add to back	O(1) amortized	O(1)
add at index	O(n)	O(n)
remove front	O(n)	O(1)
back	O(1) amortized	O(1)
index	O(n)	O(n)
get at index	O(1)	O(n)

MAP/DICTIONARY
(and Set)

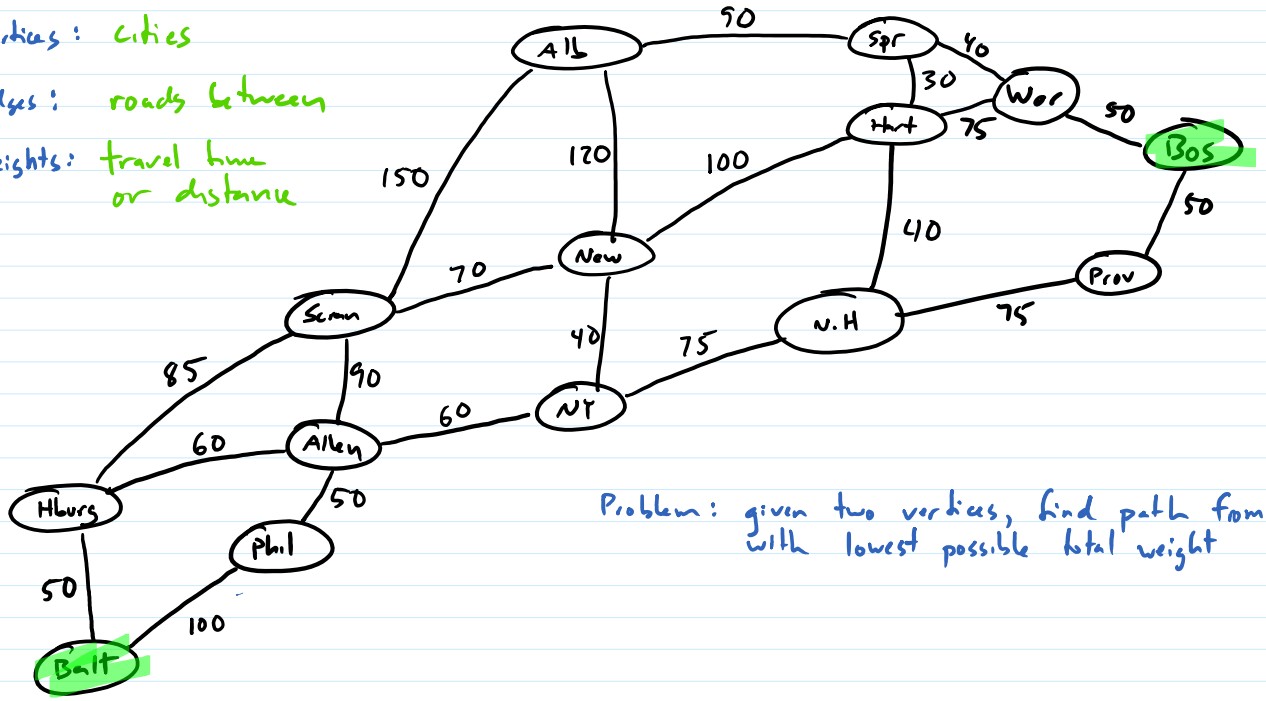
implementations

operations	hash tables	balanced BST
put	↑	↑
get	O(1) expected	O(log n)
containsKey	O(n) worst case	
	↓	↓

vertices: cities

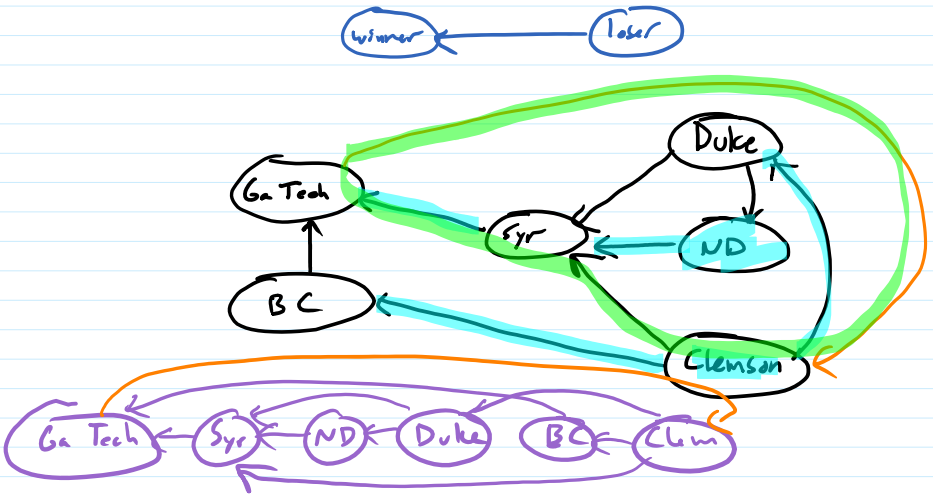
edges: roads between

weights: travel time or distance

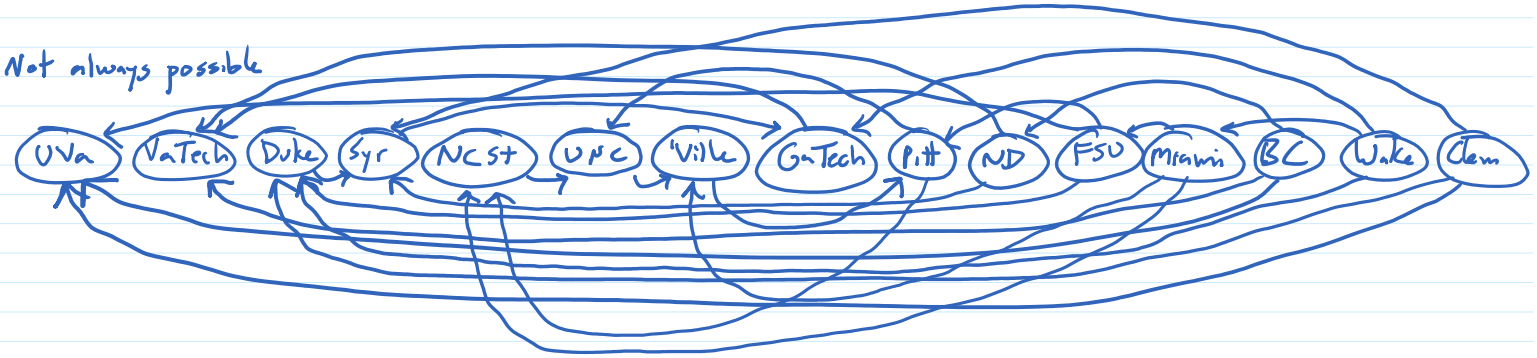


Problem: given two vertices, find path from one to other with lowest possible total weight

Feedback Arc Set



Order vertices so all edges go ←



find ordering of vertices to minimize "wrong way" edges

Brute Force: try every ordering — $n!$ — $n \geq 350$ infeasible
for each, compute number of upsets
keep track of running minimum

Feedback Arc Set is NP-complete set of problems that are hardest in their class
no one has found poly-time g's
no one has proved one doesn't exist