

EXAM 3 REVIEW!Counting Sort

How it Works

Pos<sup>①</sup>, Anew lists and what they're used for / how they interrelate  
pseudocodeTime:  $\Theta(n+k)$  w/  $k = \max(\text{integers } 0 - \text{max inclusive})$ Av x:  $\Theta(n+k)$  ↑ possible k dep. on n.

Stable: Yes!

In-Place: No!

Radix Sort sort by sort by

How it works: rightmost → leftmost.

Time:  $\Theta(d f)$  where  $f = \text{underlying sort}$ . Typically C.S. so  $\Theta(d(n+k))$   
possible d dep. on n.

d = # digits!

Stable: Yes.

in-place: No if C.S. used

Karatsuba

How the alg developed.

Reducing SDMs.

Led to  $T(n) = 3T(\frac{n}{2}) + \Theta(n) + \text{Master Thm.}$ 

How could other splittings lead to other rec. relns? M.Thm there!

Tree diagram to count SDM.

Graphs~~Walks, trails, paths~~

Cycles, trees, connected graph

~~loops, multiple edges~~

directed, weighted

Adj matrix, Adj list

Shortest Path Alg

Adj matrix, Adj list

### Shortest Path Alg

Finding SP from some  $s$  to some  $t$  or to every vertex

For unweighted, undirected, ..., graph

works from  $s$  outwards in a BF way.

Time  $\Theta(V^2)$  Adj mat

uses a queue, dist, pred lists.

$\Theta(V+E)$  Adj list.

path reconstructed from pred

run alg by hand - looking for  $t$

- not looking for  $t$

- until some instance

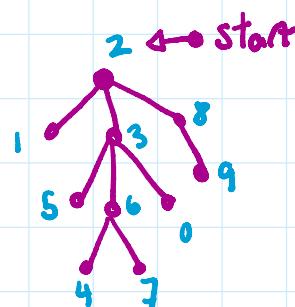
} what do the lists  
look like?

pseudocode

### BFT + DFT

BFT stays close to start as much as poss

DFT goes deep until it must back up.



DFT: 2, 1, 3, 0, 5, 6, 4, 7, 8, 9

BFT: 2, 1, 3, 8, 0, 5, 6, 9, 4, 7

inc order      inc order      inc order  
blk of 6      blk of 8      inc. order  
blk of 3      inc.order      inc.order

### BFT

How it works!

Queue, visited, vorder

I would give pseudocode

run alg by hand

Tc:  $\Theta(V+E)$  adj list.

### DFT

How it works.

- Stack, visited, vorder.  $\leftarrow$  not  $\Theta(V+E)$ ! its  $\Theta(V^2)$ !
  - DLL/stack, visited, vorder.  $\leftarrow \Theta(V+E)$ !
  - run alg by hand
- } w/ adj list!

## Dijkstra

Finds min weight path from  $s$  to all other vertices.

think: weighted graph version of SPA.

Set of visited vertices, dist, predecessors.

Pseudocode

run alg by hand

$\Theta(v^2 + E)$  w/ adj. list.

## Floyd

Finding least weight path b/w each pair of vert  
weighted, directed graph. (1-indexed)

Dynamiz programming:

Starts w/  $d = \text{adj. matrix} = \text{SP}$ . if no intermediate vert. allowed.

Update to allow int. vertex 1

" " " " " 2

:

until we allow them all!

Fill in the "pass by" matrices directly from graph (as in class)

No: Using one iteration of  $d$  to get the next (lot of work!)

What is  $P$  and how does it update as  $d$  updates?

Using final  $P$  to reconstruct the path for two specific vertices.

Tc:  $\Theta(v^3)$  w/ adj. matrix.

Hw: if  $P$  is given: blank in  $p[i,j]$  means there is no pred  
of  $j$  in a SP  $i \rightarrow j$

easy: you don't need it! ☺