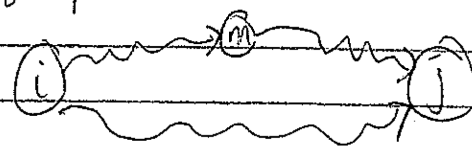
Let's # the V's. 1 to n.

We need to figure out what other ways we can specify a choice among options.

consider



can consider the paths that goes from i to j using no vertex w/ index $\geq m$.
bottom path or path that goes through m.

~~this~~ is $i \rightsquigarrow m$ & $m \rightsquigarrow j$ each have intermed. v's that are all $\leq m-1$.
also $i \rightsquigarrow j$ has intermed. v's $\leq m-1$.

So ~~we can either choose~~ to go

from i to j using no vertex w/ # $> m$

we can choose either to

go ~~from~~ through m or go through v's all $\leq m-1$.

if we go through m we have 2 subproblems $i \rightsquigarrow m$ & $m \rightsquigarrow j$.
if not, we have 1 subproblem $i \rightsquigarrow j$ w/all $\leq m$

Choice	Sub. Prob(s)	Thm. Cost
go thru m	i, m, m-1 m, j, m-1	0
don't go thr. m	i, j, m-1	0

Value Function

$$F(i,j,m) = \text{weight of SP. from } i \text{ to } j \text{ using no intermed. v\# } > m.$$

recursive:

$$F(i,j,m) = \min\big(F(i,m,m-1) + F(m,j,m-1),\ F(i,j,m-1)\big)$$

BASE CASE go through no intermed. v's.

$$F(i,j,0) \leftarrow w_{ij}$$

call this $F_{ij}^{(m)}$ call this $F_{ij}^{(0)}$.

FLOYD-WARSHALL(W, n) $n = |V| = $ dims of W.

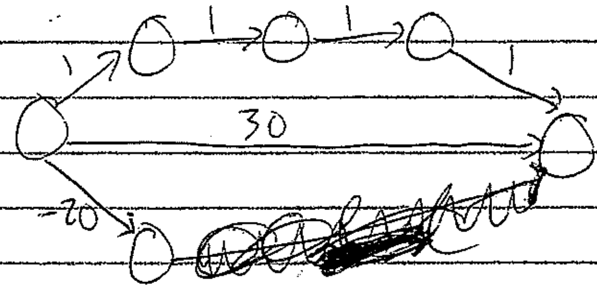
```
F^(0) <- W
for m <- 1 to n
    for i <- 1 to n
        for j <- 1 to n
            F_ij^(m) <- min(F_im^(m-1) + F_mj^(m-1), F_ij^(m-1))
return F^(n)
```

$\Theta(n^3)$.

FLOYD-WARSHALL	$\Theta(V^3)$	really small contants
Rep. Squaring	$\Theta(V^3 \lg V)$	
Rep. Dijkstra's	$\Theta(V^2 \lg V + VE)$	– no neg. wts.
" B-F	$\Theta(V^2 E)$.	

If we had sparse graph which do you prefer?
We want to do Repeated Dijkstra's but w/ NO NEG WTS.

e.g.



can we just change the weights in some way?

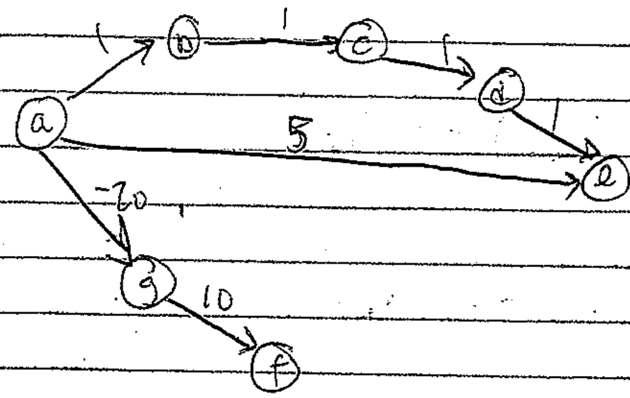
add 20 to all? ~~best is thing bottom 2 edges~~ but.

$4 \cdot 21 = 84$ worse than $20+30=50$.

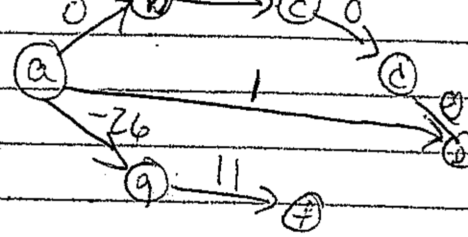
SP's w/ more edges are penalized too much.

~~[illegible]~~

~~START w/ new notation etc. on pg 252 258~~



e.g. a = 1 $\quad h(a) = 1$
b = 2 $\quad h(b) = 2$ etc.
etc.



NUMBER THE V'S. w/ any #'s (RANDOM ORDER). consider these #'s as $h(v_i)$

$1 \xrightarrow{10} 2$ replace weight w/ $1-2+10=9$

$10 \xrightarrow{5} 20$ replace wt. w/ $10-20+5=-5$.

WE want to add a rule to make sure they are non-negative but changing the weights in this way does not alter the shortest paths.

NOTICE: $v_1 \longrightarrow v_2 \longrightarrow v_3 \longrightarrow \cdots \longrightarrow v_k$

the path wt is $w(v_1,v_2)+w(v_2,v_3)+\cdots+w(k-1,k)$.

263

$$w(V_1, V_2) + h(V_1) - h(V_2)$$
$$+ w(V_2, V_3) + h(V_2) - h(V_3)$$
$$+ w(V_3, V_4) + h(V_3) - h(V_4)$$

$$\vdots$$

$$+ w(V_{K-1}, V_K) + h(V_{K-1}) - h(V_K).$$

$$= w(V_1, V_2) + h(V_1) - h(V_K)$$
$$+ w(V_2, V_3)$$

this is the only difference

$$\vdots$$

$$+ w(V_{K-1}, V_K)$$

ANY PATH THAT GOES FROM V_1 to V_K has that $h(V_1) - h(V_K)$ additional value added to it. so, If you add the same # to all numbers, the order doesn't change. and because $h(V_1) - h(V_K)$ does not depend on the PATH therefore it cannot screw up the shortest path values. or any of the better paths ~~from~~ before the same two v's

∮

eq.

THE REASSIGNING OF THE WEIGHTS DOES NOT CHANGE WHICH PATH IS SHORTEST.

THE ONLY OTHER THING TO DO IS TO NOW MAKE ~~[illegible]~~ IT SO THAT NONE OF THE WEIGHT ARE NEGATIVE (SO WE CAN USE ~~[illegible]~~ SOME FORM OF DIJKSTRA'S)

WE WANT $w(u,v) + h(u) - h(v) \geq 0$.
that is, the new reassigned weight must be non-negative

we want $h(v) \leq w(u,v) + h(u)$.

NOTICE THIS IS LIKE THE TRIANGLE INEQUALITY.

NOW THE BIG STEP.

SUPPOSE WE HAVE distances FROM SOME SOURCE s.

$$\mathrm{dist}(s,v) \leq w(u,v) + \mathrm{dist}(s,u).$$

assign $\mathrm{dist}(s,v)$ to $h(v)$ $\mathrm{dist}(s,u)$ to $h(u)$. etc. SO WE CAN ADD A NEW Vertex s to the graph. w/ a directed edge from s to each of the other v's w/ weight 0 $w(s,u) \leftarrow 0 \; \forall u \in V$.

267

Reminder of: want to do rep. Dijkstra's b/c of sparse graphs - but want to remove neg. wts. if exist. but need to make sure not to screw up S.P's. 2 ideas. if replace V's w/ H's & ①—② $V_1 - V_2 + w_{12} = w'_{12}$

THIS IS THE INITIAL STEP ~~THE~~ THE ALGORITHM.

AND REQUIRE $w + h(u) - h(v) \geq 0$

JOHNSON (V, E, w)

1. Create a new source vertex s (already not in V) w/ $w(s,u) \leftarrow 0$ $\forall u \in V$. $V' = V \cup \{s\}$ E' is E w/ new edges.

$h(v) \leq w + h(u)$

2. RUN BELLMAN-FORD(V', E', s).
   TO COMPUTE $d[u]$ $\forall u \in V$ which are the S.P's from s to each v

$\Downarrow$ $dist(s,v) \leq w + dist(s,u)$

3. IF NWC from BELLMAN-FORD RETURN NWC.

$\Downarrow$

4. for each edge (u,v)
5. new $w(u,v) \leftarrow w(u,v) + d[u] - d[v]$
6. RUN DIJKSTRA'S ON EACH vertex as a source.
7. for each pair of v's u,v
   final $d(u,v) \leftarrow$ new $d(u,v) - d[u] + d[v]$

from dijkstra's.

add new s w/ 0 wts edges from it to all v

run BF(s).

then replace wts.

Retc.

USE THIS FOR SPARSE GRAPH W/ NEG. WT. EDGES.

Line 1 runs in V time

2 $\Theta(VE)$

4/5 E time.

6. $\Theta(V^2 \lg V + VE)$

7 V^2

$\Theta(V^2 \lg V + VE)$. - JOHNSON's

ALL PAIRS S.P.

REPEATED SQUARING $\Theta(V^3 \lg V)$

FLOYD-WARSHALL $\Theta(V^3)$ small constants

REPEATED - BELLMAN-FORD $\Theta(V^2 E)$

" DIJKSTRA'S $\Theta(V^2 \lg V + VE)$ w/ Fib. Heap.

requires NON-NEG. WTS.

JOHNSON $\Theta(V^2 \lg V + VE)$ allows. NEG. WTS.