

Grafici pesati

$$w: E \rightarrow \mathbb{R}^+$$

Comminimi di asprante
singolo Δ

Q : coda di priorità rispettando
le etichette $el(n)$: $UveV$

$$el(n) = \Delta(\Delta, n)$$

qualora n è estratto
dalla coda di priorità

Nuova operazione per

tri-structura

$$\begin{array}{ccc} u & & v \\ \downarrow & \rightarrow & \downarrow \\ \emptyset & & \emptyset \end{array}$$

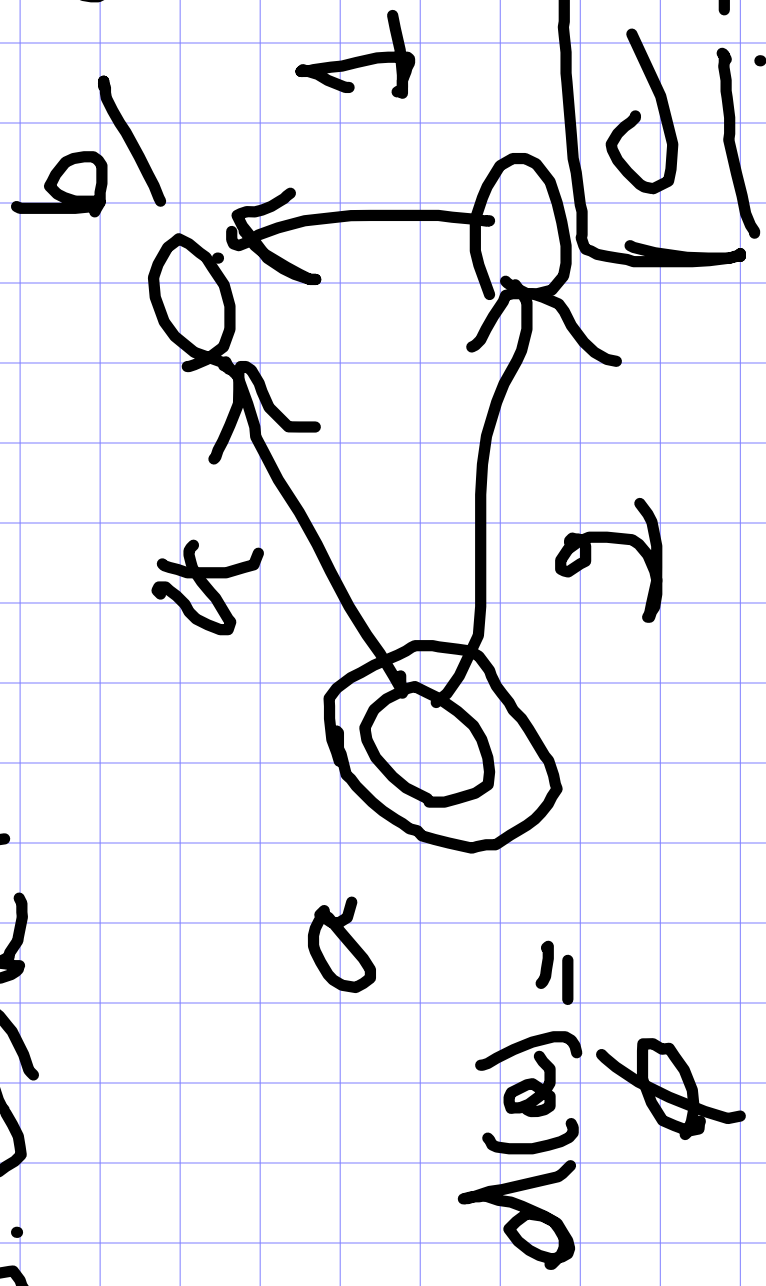
$\xrightarrow{u} \underbrace{d(u) + w(u, v)}$

Relax $(u, v, w(u, v))$

if $\boxed{d(v)} \rightarrow \boxed{d(u) + w(u, v)}$ then

$$\begin{cases} d(v) \leftarrow d(u) + w(u, v) \\ \pi(v) \leftarrow u \end{cases}$$

$w: E \rightarrow \mathbb{R} + \text{source} = a$



$$d(a) = \emptyset$$

$$d(a) = \underbrace{\emptyset(a, a)} = \emptyset$$

$$d(b) = +\infty$$

$$d(b) = d(a) + \cancel{w(a, b)}$$

$$\cancel{4} + \cancel{3} = 0$$

$$d(c) = +\infty$$

$$d(c) = a \quad \pi(c) = a$$

$$a \quad b \quad a = b$$

$$d(b) = 4 > d(c) + w(c, b)$$

$$2 + 1$$

$$\{d(b) = 3$$

$$\pi(b) = c$$

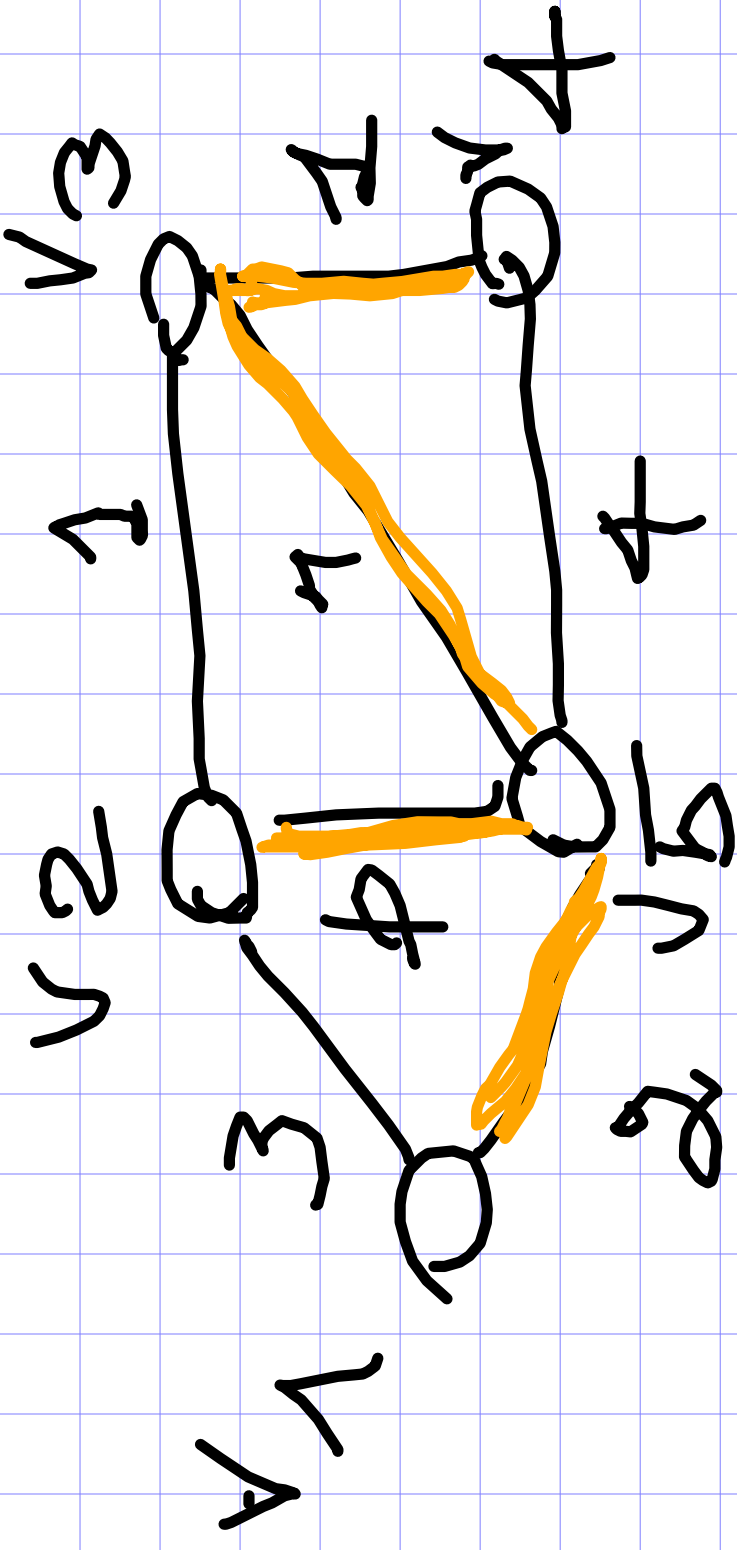
Dijkstra ($G, w: E \rightarrow R^+, s$)
{ $\forall v \in V: \text{color}(v) = \text{white}$
 $d(v) = \infty; \pi(v) \leftarrow \text{nil}$

$d(s) \leftarrow 0; \text{Sol} \leftarrow \emptyset$

$Q \leftarrow \text{QUEUE}(G, d(v): \forall v \in V)$

 while Q is not empty do

$u \leftarrow \text{extract} \cdot \min(Q)$
 $\forall v \in \text{Adj}(u)$ and $v \notin \text{Sol}$
 $\text{RELAX}(u, v, w(u, v))$
 $\text{Sol} \leftarrow \text{Sol} \cup \{u\}$
 }



π	v_1	v_2	v_3	v_4	v_5
d	∞	2	3	4	2

v_1 v_2 v_3 v_4 v_5
 \uparrow \uparrow \uparrow \uparrow \uparrow

∞	2	3	4	2
v_1	v_2	v_3	v_4	v_5

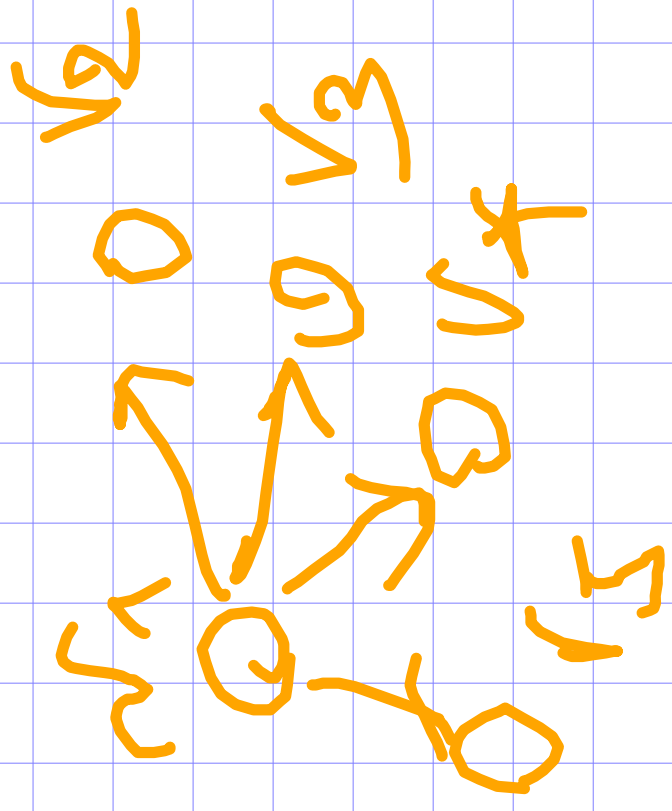
$S = v_1$
 $Sol \leftarrow \{v_1, v_5, v_2, v_3, v_4\}$

$relax(v, v_5, w(v, v_5))$
 $[d(v_5) > \infty(1)]$
 $d(v_5) \leftarrow w(v, v_5)$
 $d(v_5) \leftarrow d(v) + w(v, v_5)$
 $\pi(v_5) \leftarrow v$

Complessità

$|V|$ • $\text{extract} - \min$

+ $|E|$ • relax



Coda Q realizzata de un vettore

$\text{extract} - \min \quad O(|Q|) = O(|V|)$

relax : accelero diretto a Q

usando l'indice
dell'offset $O(1)$

Computer to' can code di fronte: realizzo
con vettore

$$|V| \cdot \text{Extract-min} - |E| T_{\text{relax}}$$

$$= |V| \cdot O(\log V) + |E| O(1)$$

$$= O(V^2 \log V) + O(|E|) = \boxed{O(V^2)}$$

$$E = V^2$$

ve: codice

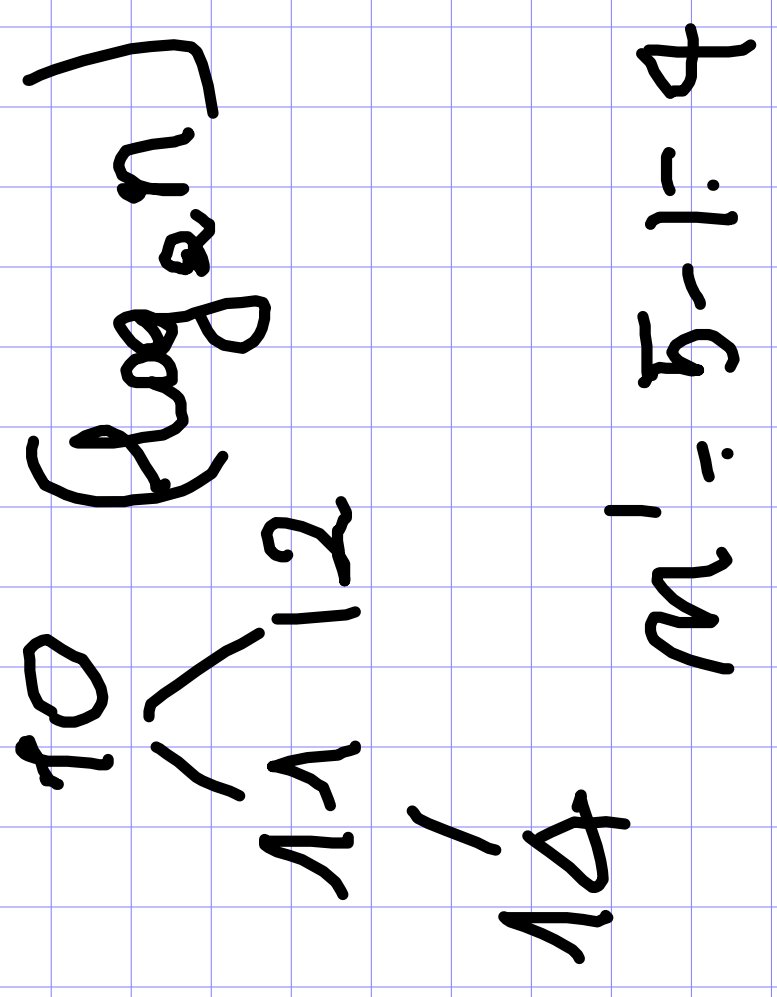
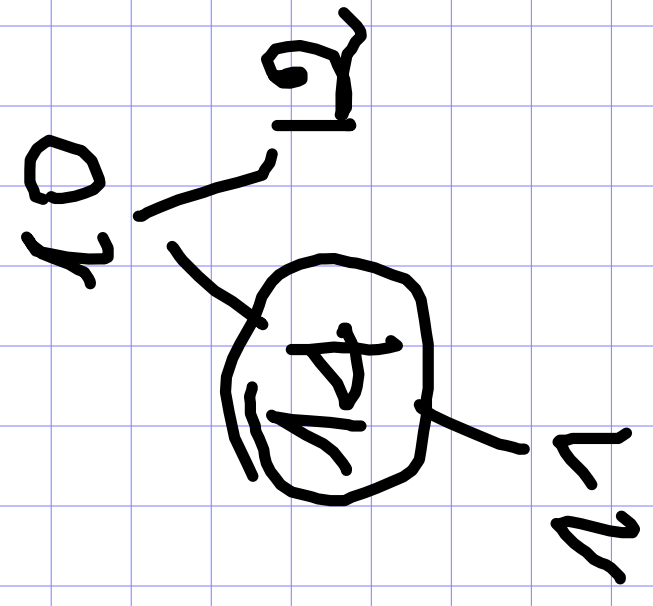
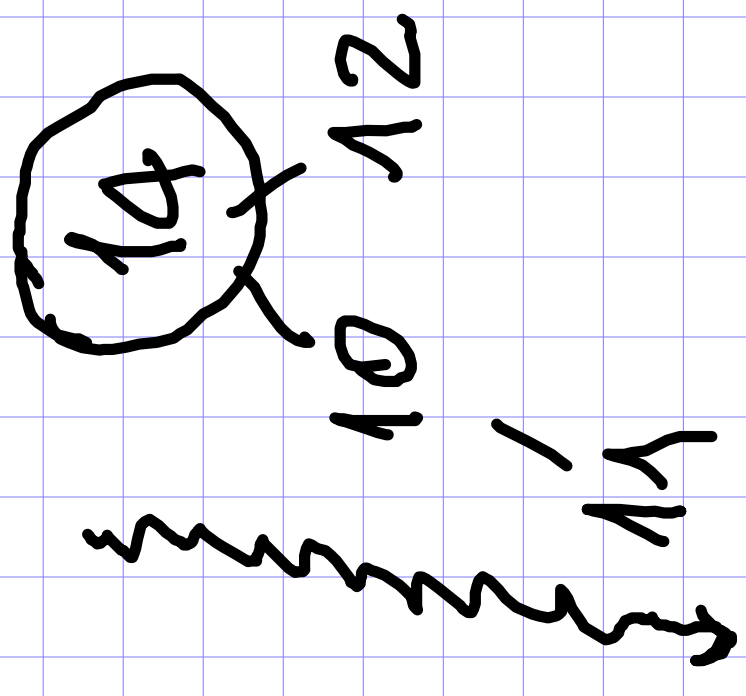
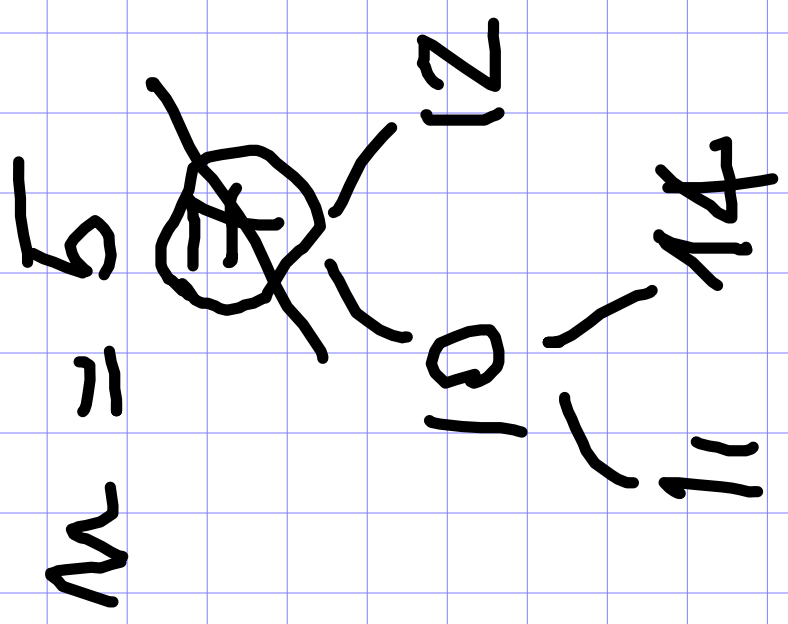
2. A $E = \text{numero}$ anal: nel
grafo

\Rightarrow Struttura non in fluisce

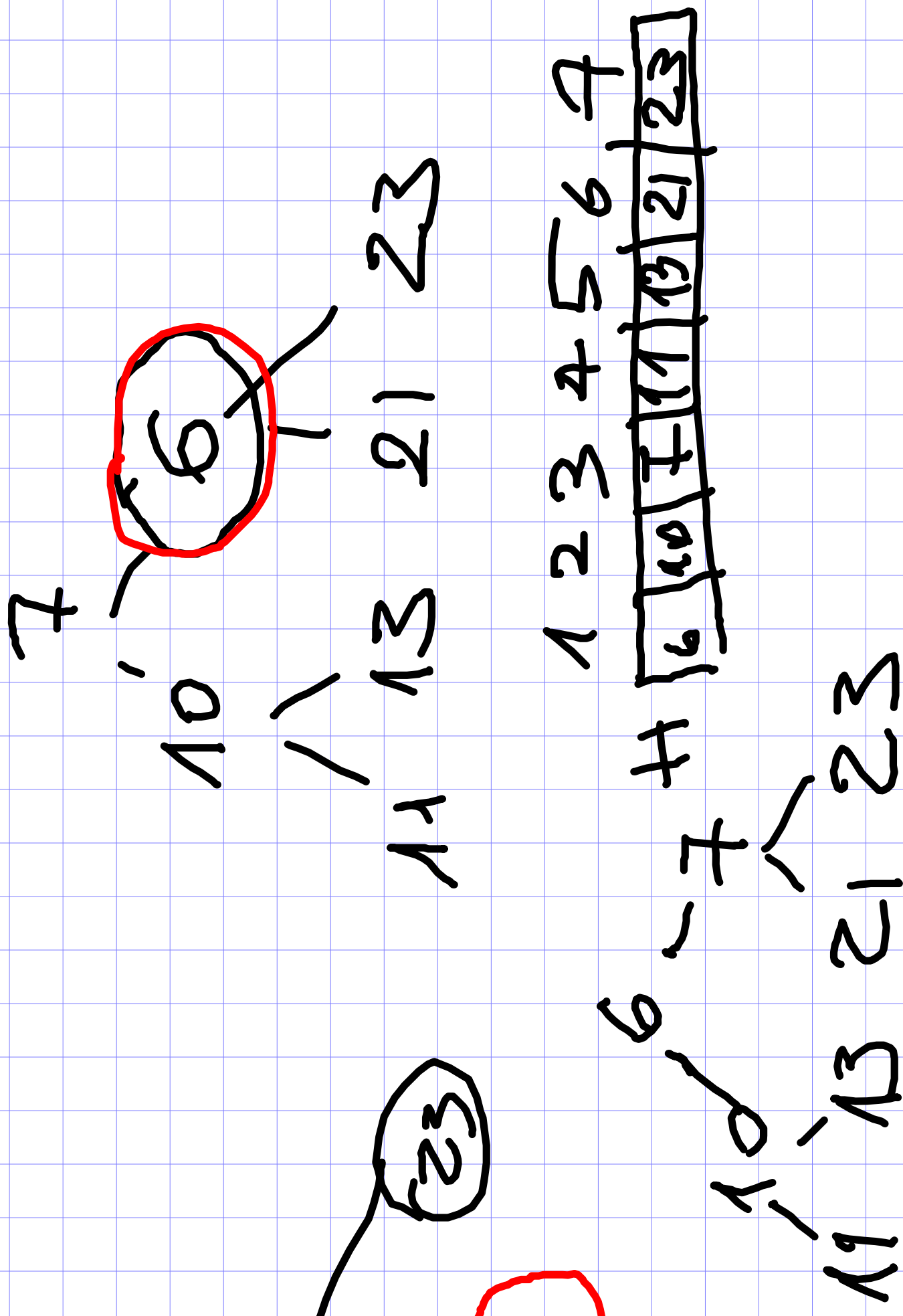
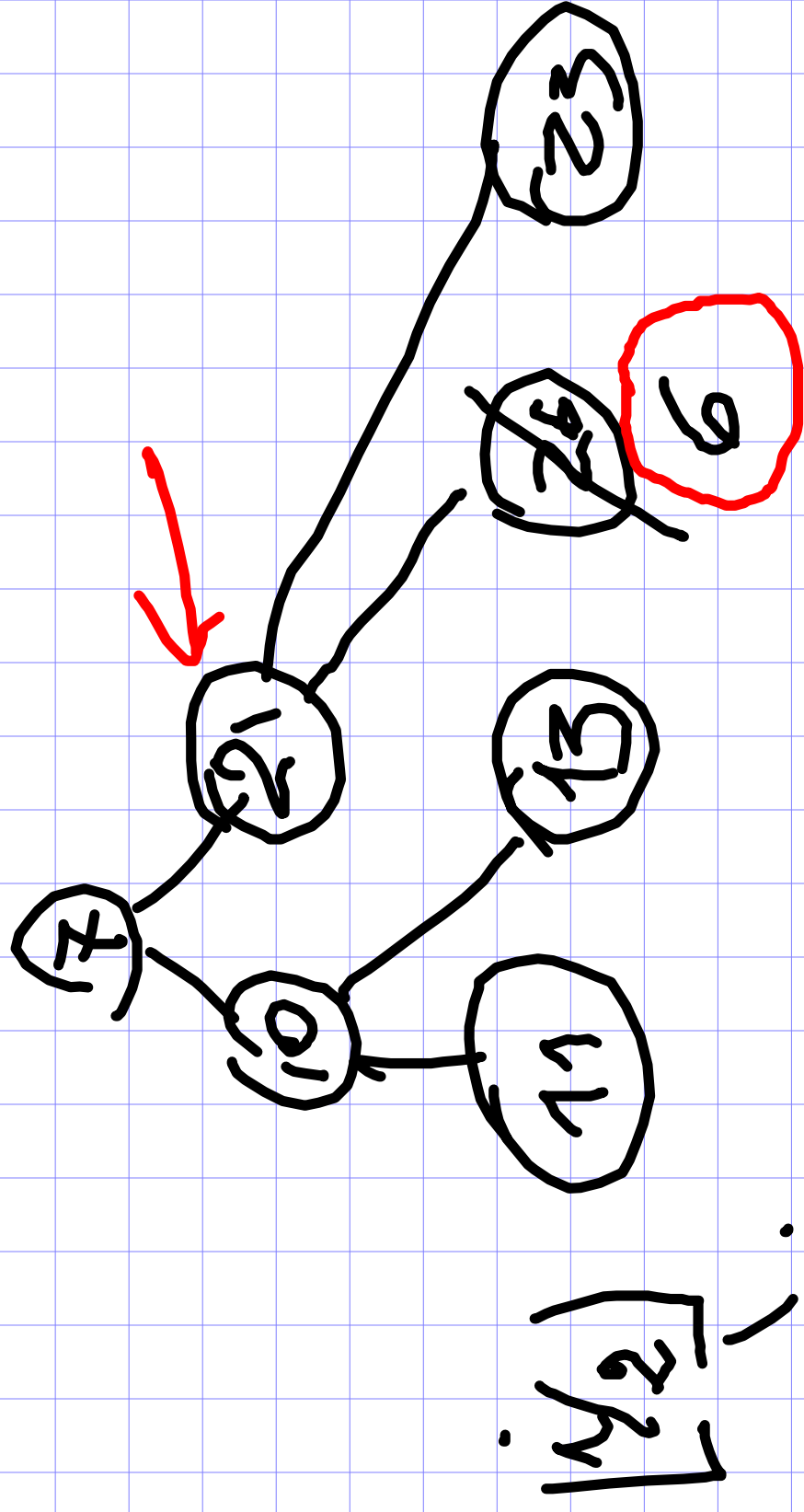
- $\text{extract-min} = \text{return } tt(i)$
 $ttcapify(i, m-1)$

- $\text{RELAX} = \text{decrease-key}$
 \rightarrow technique to remove
vertex from priority

Example extract-min the heap binario



Esempio di un min-heap



// $key \leftarrow H[i] //$

Decrease $\leftarrow key (H, i, key)$

$H[i] \leftarrow key;$

while

$(i > 1)$

and

$H[i] < H[\lfloor i/2 \rfloor]$

do { combine $(H[i], H[\lfloor i/2 \rfloor])$ }

$\text{max} = O(\log n) \quad n = |V|$

Dijkstra $(G, w: E \rightarrow \mathbb{R}^+, s)$

$\forall v \in V: \text{color}(v) = w; \pi(v) \leftarrow \text{nil}$

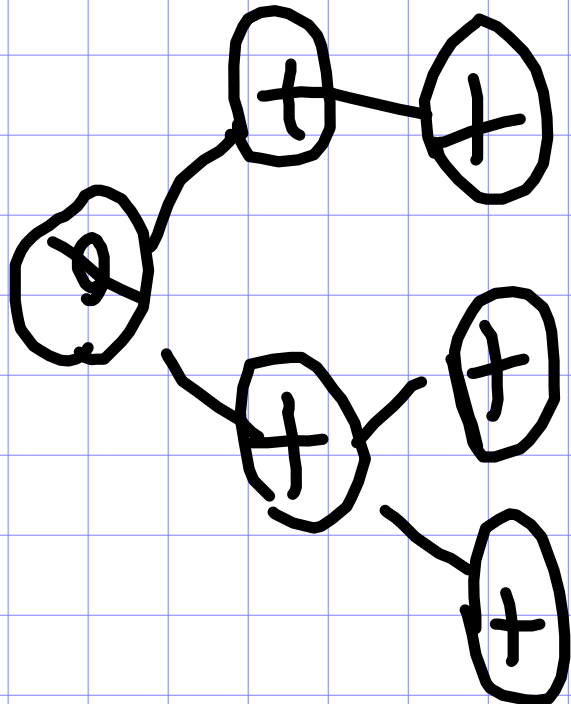
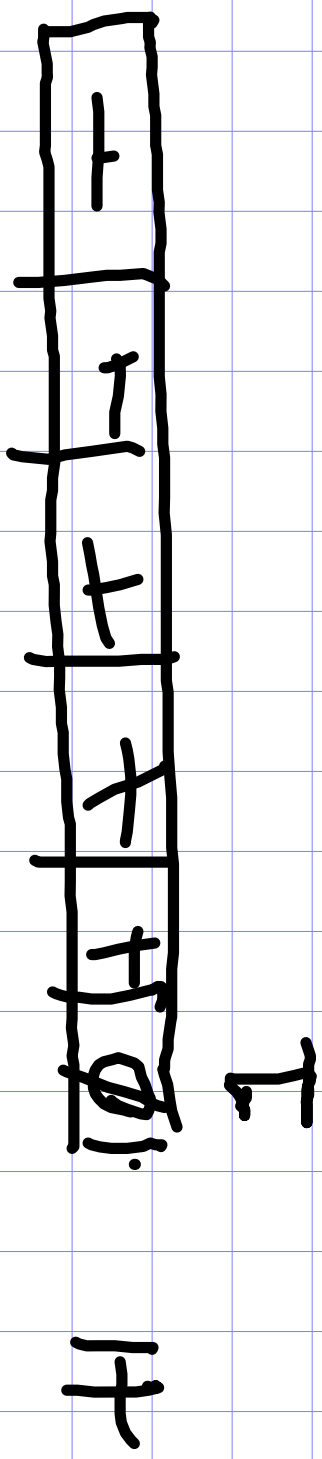
$d(s) \leftarrow 0$

$H \leftarrow \text{MAX-HEAP}(d(v): \forall v \in V)$

while H is not empty do

$u \leftarrow \text{extract-min}(H)$
 $\forall v \in \text{Adj}(u)$ and $v \notin \text{sol}$
HEAP-RELAX($u, v, w(u, v)$)
 $\text{sol} \leftarrow \text{sol} \cup \{u\}$

maximize A (of n) : $\forall n \in \mathbb{N}$



Decrement key ($\#$, i , key)

$\# [i] \leftarrow \text{key};$

while

$(i > 1)$

and

$\# [i] < \# [i/2]$

do { combine

$(\# [i], \# [i/2])$

}

(HEAP): RELAX($u, n, w(u, n)$)

if $d(n) > d(u) + w(u, n)$

then

$\rightarrow d(n) \leftarrow d(u) + w(u, n)$

$\pi(n) \leftarrow u$

DECREASE-KEY(π , $\underbrace{pos(n), d(n)}$)

// maintain h as a min-heap // all: u is internal

Complexity

$|V| T_{\text{extract}} - \min$

$|E| T_{\text{RELAX}}$

$(|V| + |E|) (\log_2 |V|)$

code
complexity

heap binario
con

$$d(r) \leftarrow \underbrace{d(u) + w(u, r)}$$

DECREASE-key ($t, \underbrace{f(u, r)}_{\text{mode}}, \underbrace{d(r)}_{\text{key}}$)

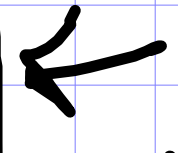
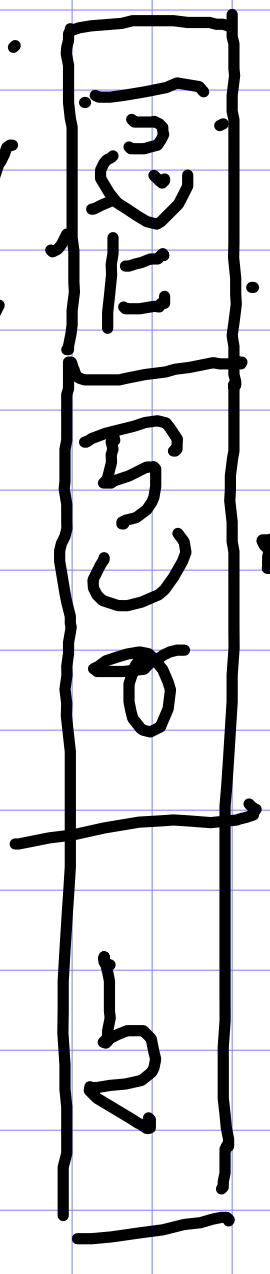
Seus montenera. we correspond
 buvivo tra i verticiz-e le
 low parione in t (del gofo)

Quanto segue è l'implementazione
di Dijkstra con heap.
È necessario stabilire una
corrispondenza tra i nodi
e i vertici del graf e i nodi
dello heap.

di $n \in V$

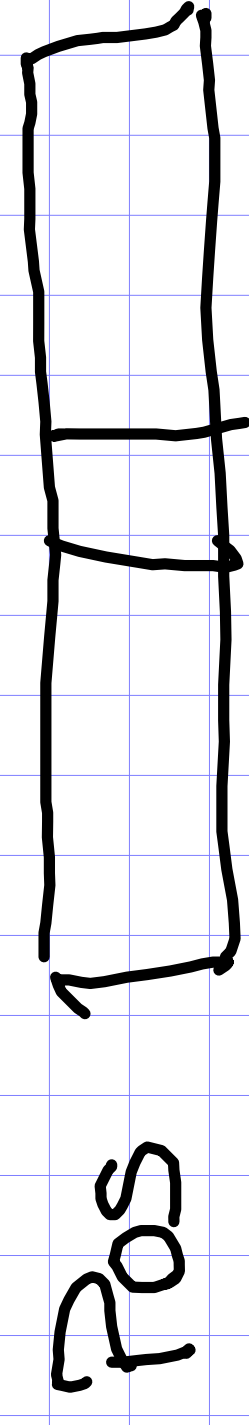
Il memoria π a
in posizione

Moio dello heap



Il memoria a
in posizione

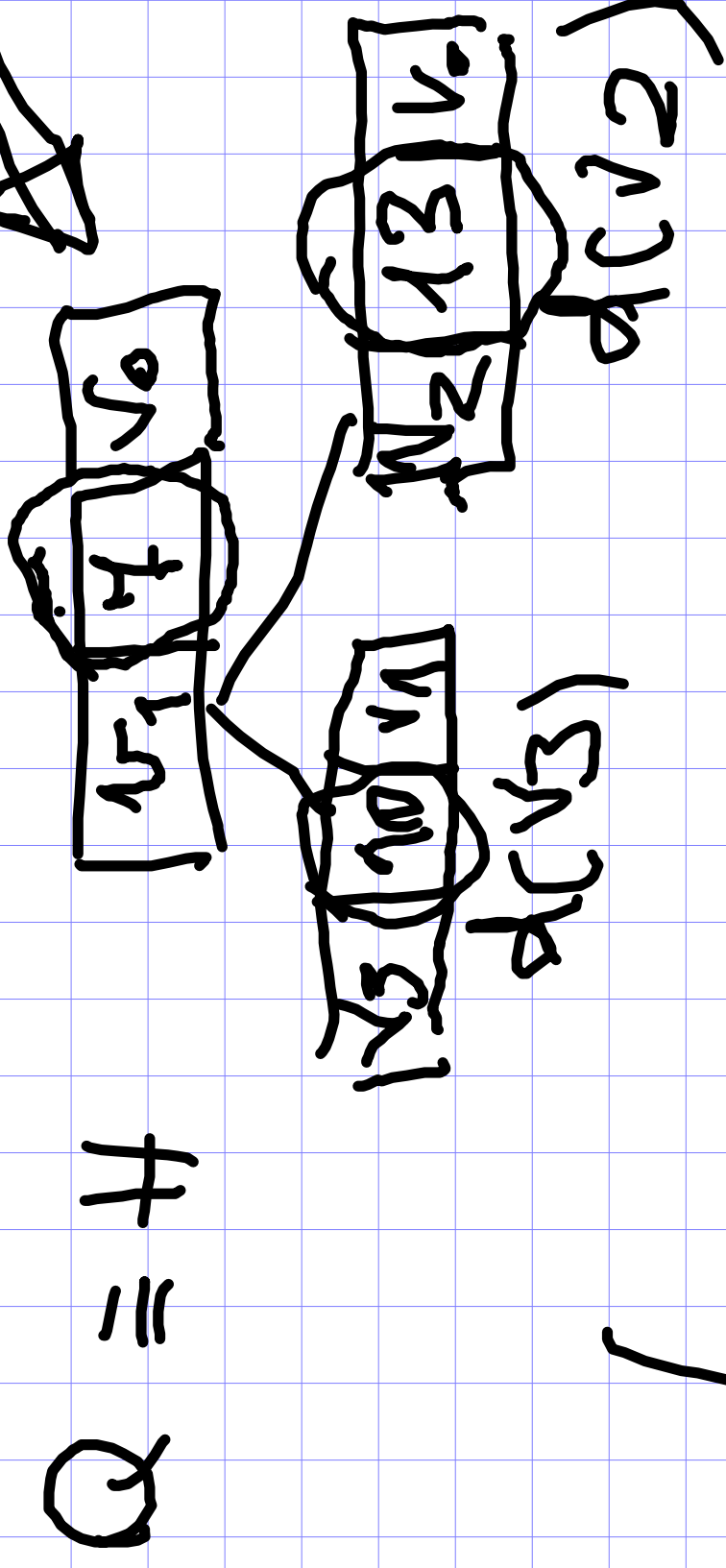
Pos: vettore per trovare la posizione
di n in π in $O(1)$



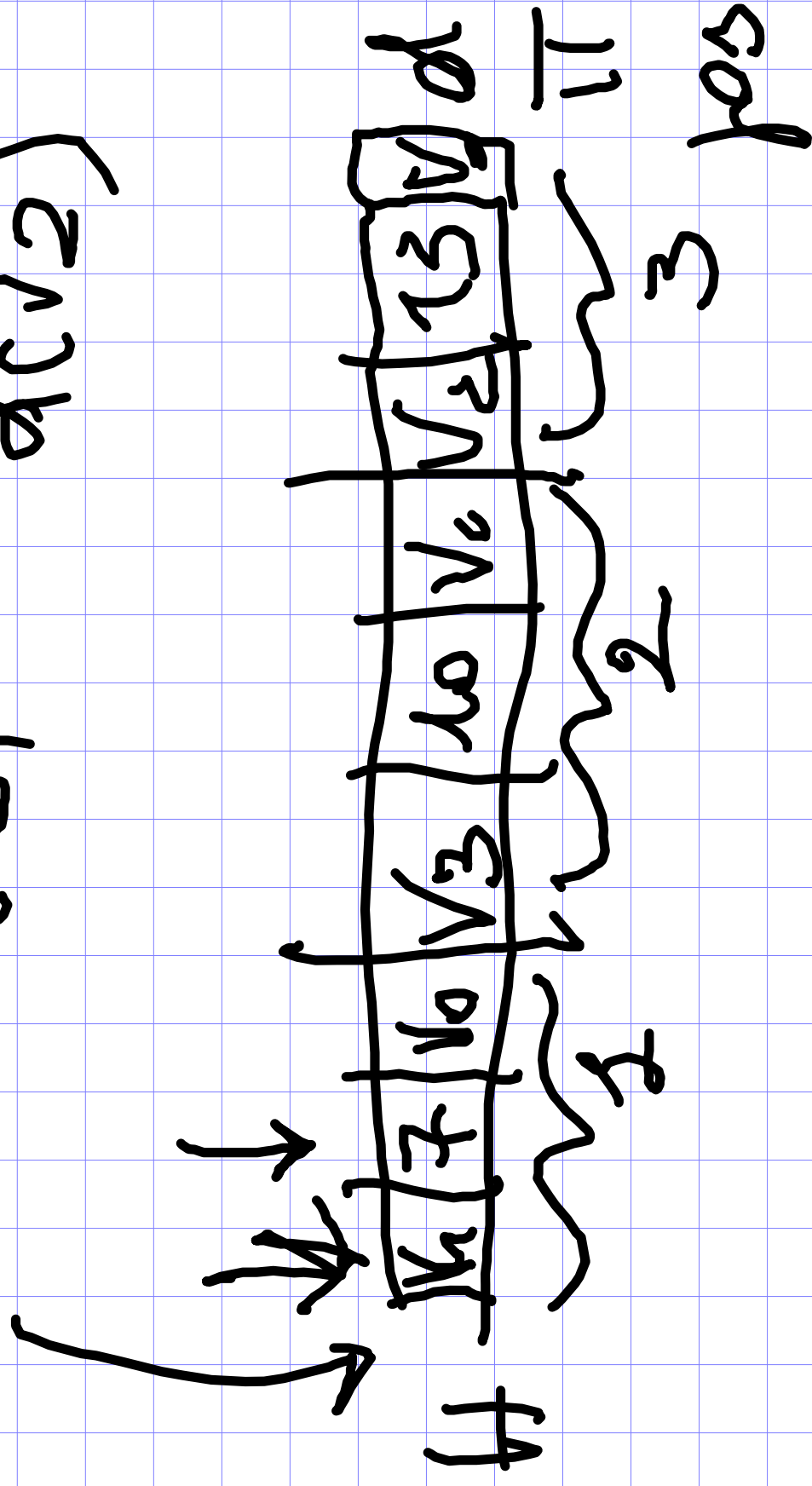
$Pos(n) =$ posizione di $d(n)$
value del graph in π

Exemplo

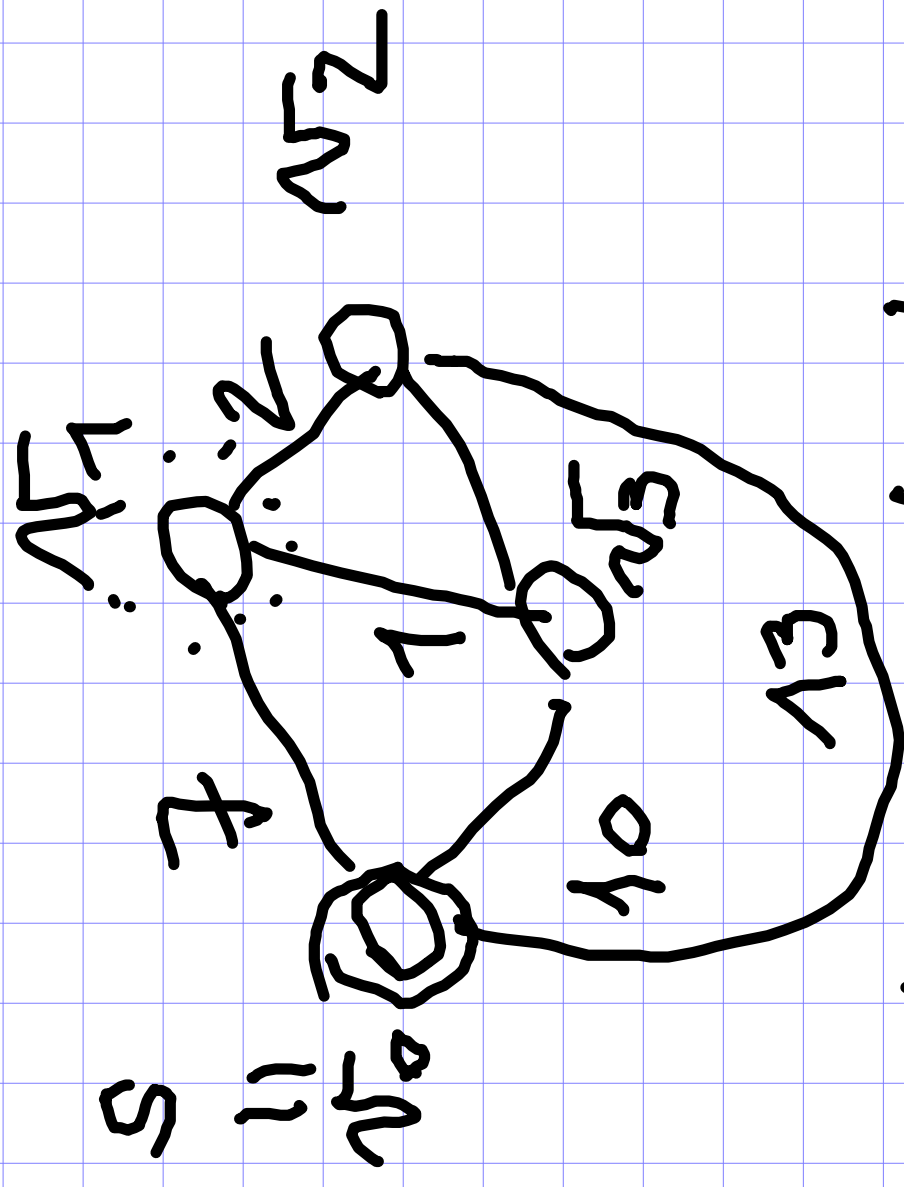
Example



≠
|||
d



50
100

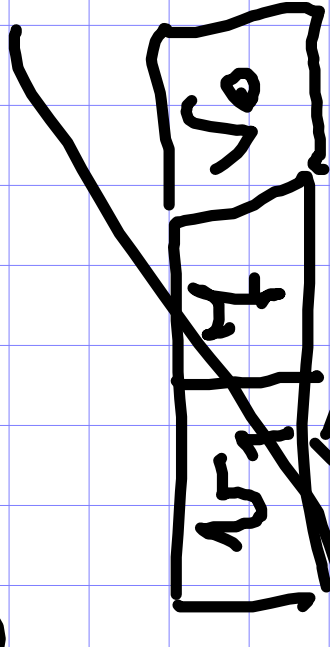


v_0	v_1	v_2	v_3
0	7	13	10
nie	v_0	v_0	v_0
1	1	3	2

$u \leftarrow N_1$ (extract-min)
 $DELA\gamma\{v_1, v_3, 1\}$

Example

$Q = H$



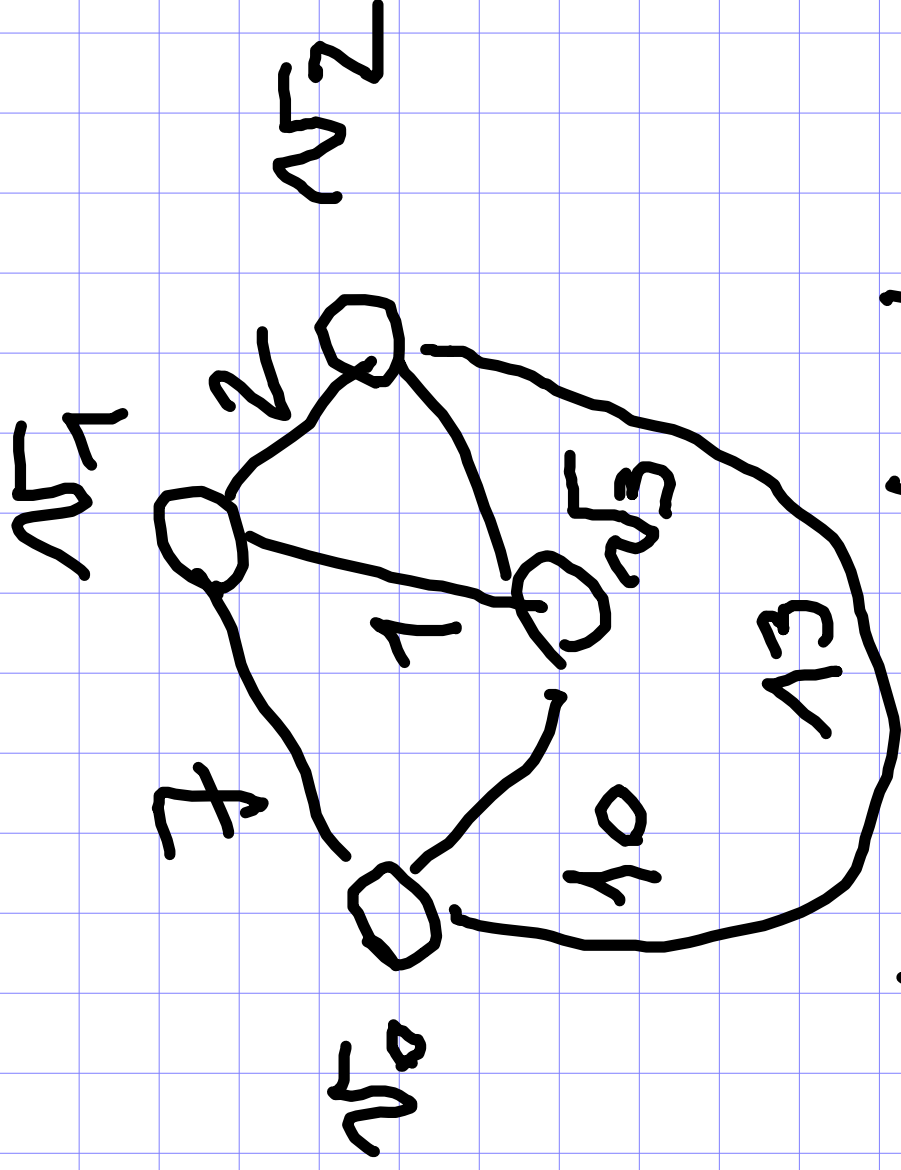
nodes d H g



10 10 10
 extract-min

H H

d H g
 11 11 10



v_0	v_1	v_2	v_3
0	7	13	10
min	v_0	v_0	v_0
$+$	$-$	0	$+$

The permutations can maping g to h - leaf

Decrease-key (H , key)

$H[t].ol \leftarrow \text{key};$

while ($t > 1$) and

$(H[t].ol < H[\lfloor \frac{t}{2} \rfloor].ol)$

{ $\text{temp} \leftarrow H[t]$

$H[t] \leftarrow H[\lfloor \frac{t}{2} \rfloor];$ $\text{pos}(H[t].\text{node}) \leftarrow t;$

$H[\lfloor \frac{t}{2} \rfloor] \leftarrow \text{temp};$ $\text{pos}(H[\lfloor \frac{t}{2} \rfloor].\text{node}) \leftarrow \lfloor \frac{t}{2} \rfloor$

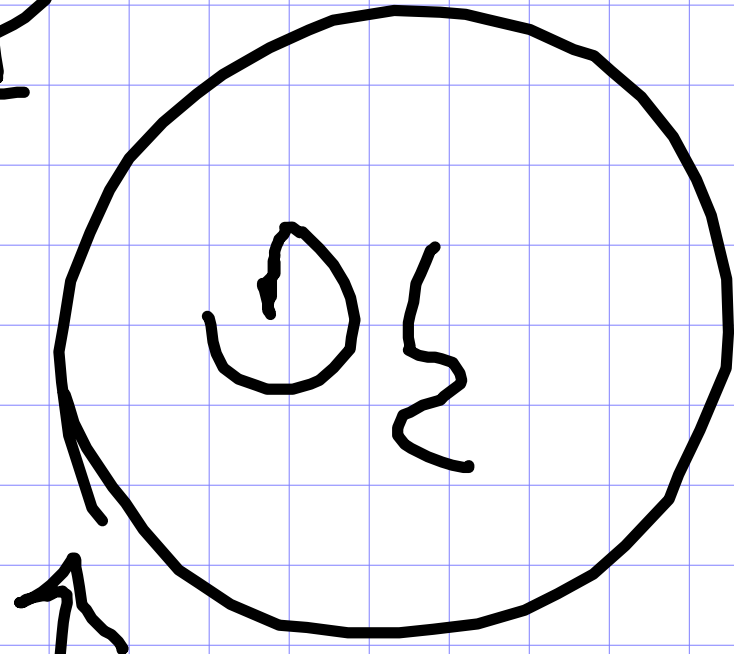
$t \leftarrow \lfloor \frac{t}{2} \rfloor$

}

Illustration of risk reduction
for vertex G - H crop

†

extract- m

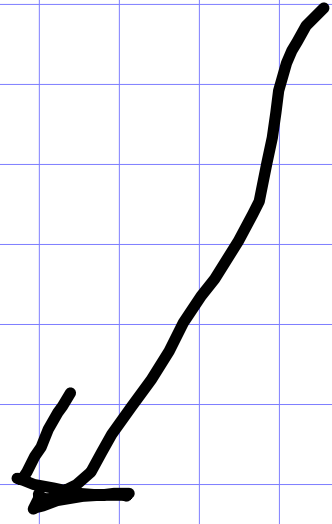


relax

$d(v)$

\rightarrow

$pos(v)$



Dijkstra $(G, w: E \rightarrow \mathbb{R}^+, s)$

$\forall v \in V: \text{color}(v) = w; \pi(v) \leftarrow \text{nil}$

$d(s) \leftarrow 0$

$H \leftarrow \text{MAXHEAP}(d(v): \forall v \in V)$

while H is not empty do

$u \leftarrow \text{extract_min}(H)$
 $\forall v \in \text{Adj}(u)$ and $v \notin \text{sol}$
RELAX($u, v, w(u, v)$)
 $\text{sol} \leftarrow \text{sol} \cup \{u\}$

Complement, con heap bijnario

+
 V. Extract - min
 F. Decrease - key

$$n \cdot O[V + F] \log V$$

Opfer von Gewalt:

Esplorazione del
V1 extact-min
grpo +
17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47/48/49/50/51/52/53/54/55/56/57/58/59/60/61/62/63/64/65/66/67/68/69/70/71/72/73/74/75/76/77/78/79/80/81/82/83/84/85/86/87/88/89/90/91/92/93/94/95/96/97/98/99/100/101/102/103/104/105/106/107/108/109/110/111/112/113/114/115/116/117/118/119/120/121/122/123/124/125/126/127/128/129/130/131/132/133/134/135/136/137/138/139/140/141/142/143/144/145/146/147/148/149/150/151/152/153/154/155/156/157/158/159/160/161/162/163/164/165/166/167/168/169/170/171/172/173/174/175/176/177/178/179/180/181/182/183/184/185/186/187/188/189/190/191/192/193/194/195/196/197/198/199/200/201/202/203/204/205/206/207/208/209/210/211/212/213/214/215/216/217/218/219/220/221/222/223/224/225/226/227/228/229/230/231/232/233/234/235/236/237/238/239/240/241/242/243/244/245/246/247/248/249/250/251/252/253/254/255/256/257/258/259/260/261/262/263/264/265/266/267/268/269/270/271/272/273/274/275/276/277/278/279/280/281/282/283/284/285/286/287/288/289/290/291/292/293/294/295/296/297/298/299/300/301/302/303/304/305/306/307/308/309/310/311/312/313/314/315/316/317/318/319/320/321/322/323/324/325/326/327/328/329/330/331/332/333/334/335/336/337/338/339/340/341/342/343/344/345/346/347/348/349/350/351/352/353/354/355/356/357/358/359/360/361/362/363/364/365/366/367/368/369/370/371/372/373/374/375/376/377/378/379/380/381/382/383/384/385/386/387/388/389/390/391/392/393/394/395/396/397/398/399/400/401/402/403/404/405/406/407/408/409/410/411/412/413/414/415/416/417/418/419/420/421/422/423/424/425/426/427/428/429/430/431/432/433/434/435/436/437/438/439/440/441/442/443/444/445/446/447/448/449/450/451/452/453/454/455/456/457/458/459/460/461/462/463/464/465/466/467/468/469/470/471/472/473/474/475/476/477/478/479/480/481/482/483/484/485/486/487/488/489/490/491/492/493/494/495/496/497/498/499/500/501/502/503/504/505/506/507/508/509/510/511/512/513/514/515/516/517/518/519/520/521/522/523/524/525/526/527/528/529/530/531/532/533/534/535/536/537/538/539/540/541/542/543/544/545/546/547/548/549/550/551/552/553/554/555/556/557/558/559/560/561/562/563/564/565/566/567/568/569/570/571/572/573/574/575/576/577/578/579/580/581/582/583/584/585/586/587/588/589/590/591/592/593/594/595/596/597/598/599/600/601/602/603/604/605/606/607/608/609/610/611/612/613/614/615/616/617/618/619/620/621/622/623/624/625/626/627/628/629/630/631/632/633/634/635/636/637/638/639/640/641/642/643/644/645/646/647/648/649/650/651/652/653/654/655/656/657/658/659/660/661/662/663/664/665/666/667/668/669/670/671/672/673/674/675/676/677/678/679/680/681/682/683/684/685/686/687/688/689/690/691/692/693/694/695/696/697/698/699/700/701/702/703/704/705/706/707/708/709/710/711/712/713/714/715/716/717/718/719/720/721/722/723/724/725/726/727/728/729/730/731/732/733/734/735/736/737/738/739/740/741/742/743/744/745/746/747/748/749/750/751/752/753/754/755/756/757/758/759/760/761/762/763/764/765/766/767/768/769/770/771/772/773/774/775/776/777/778/779/780/781/782/783/784/785/786/787/788/789/790/791/792/793/794/795/796/797/798/799/800/801/802/803/804/805/806/807/808/809/810/811/812/813/814/815/816/817/818/819/820/821/822/823/824/825/826/827/828/829/830/831/832/833/834/835/836/837/838/839/840/841/842/843/844/845/846/847/848/849/850/851/852/853/854/855/856/857/858/859/860/861/862/863/864/865/866/867/868/869/870/871/872/873/874/875/876/877/878/879/880/881/882/883/884/885/886/887/888/889/890/891/892/893/894/895/896/897/898/899/900/901/902/903/904/905/906/907/908/909/910/911/912/913/914/915/916/917/918/919/920/921/922/923/924/925/926/927/928/929/930/931/932/933/934/935/936/937/938/939/940/941/942/943/944/945/946/947/948/949/950/951/952/953/954/955/956/957/958/959/960/961/962/963/964/965/966/967/968/969/970/971/972/973/974/975/976/977/978/979/980/981/982/983/984/985/986/987/988/989/990/991/992/993/994/995/996/997/998/999/1000/1001/1002/1003/1004/1005/1006/1007/1008/1009/1010/1011/1012/1013/1014/1015/1016/1017/1018/1019/1020/1021/1022/1023/1024/1025/1026/1027/1028/1029/1030/1031/1032/1033/1034/1035/1036/1037/1038/1039/1040/1041/1042/1043/1044/1

Operazioni coinvolte:

Esplorazione del graph +
 $|V|$ extract-min + $|E|$ reex
and esistendi

Proporzionalità con $L: A$.

Tempo alg di Dijkstra \rightarrow

$$O(V + E) + O(|V| + |E|) (\log_2 |V|) \\ = O(V + E) (\log_2 V)$$

Kapreventionen von M.A. →
Temperatur: alg Dijkstra

$$O(V^2) + \underbrace{(V + \sum_k E_k) (\log_2 V)}_{\text{Extract-min}} + \text{relax}$$

$$= O[V^2 + E \log_2 V]$$

L.4: f Dijkstra con trap \rightarrow

$$(V + E)(\log V) \stackrel{\text{graf}}{=} E \log V \quad (1)$$

Dijkstra con vettore $|E| \geq |V|$

$$(V^2 + E) = V^2 \quad (2)$$

$$\& |E| \in O\left(\frac{V^2}{\log V}\right) \rightarrow (1) \text{ better } (2)$$

$$\& |E| \in \Theta(V^2) \rightarrow (2) \text{ better } (1)$$

M.A + Dijkstra con heap

$$O[V^2 + E \log_z V] \quad (1)$$

Dijkstra con vector

$$O(V^2) \quad (2)$$

(2) battle (1) senza

È l'algoritmo di Dijkstra per
il calcolo dei cammini minimi
che rappresenta meglio e
conviene usare lo heap binario se
il grafo è rappresentato con la
lista delle adiacenze. $|E| \in O\left(\frac{V^2}{\log V}\right)$

ESPOSIZIONE!

Alg Dif Kshia hō m gliorare la
ma co mpleta' rafforzato solo il trlo
con liste delli Adiacere e hrap

di fibonocci (struttura ammorhiata).

Tenat - min

$$O(\log V)$$

$$O(1)$$

Talox

Completo

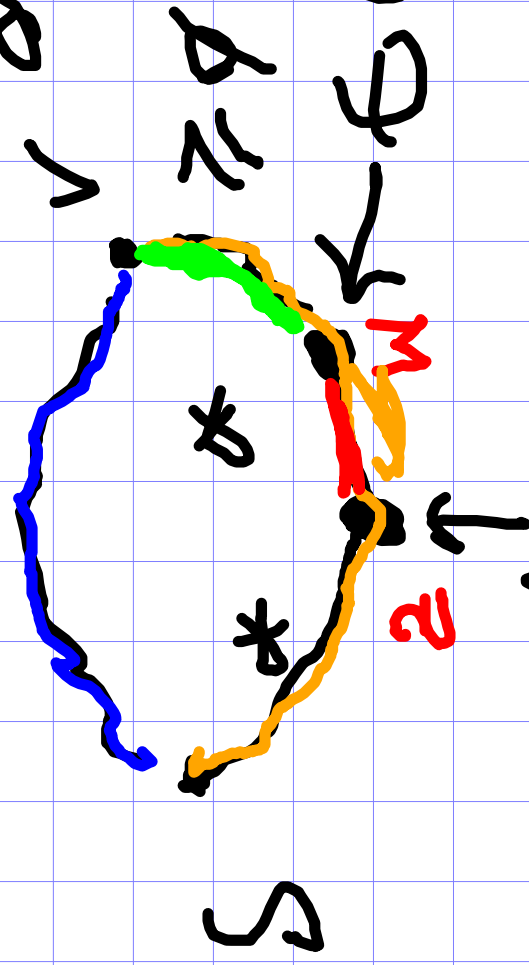
$$O(V \log V + E)$$

Correttore dell'alg. di Dijkstra
 $\forall v \in V: d(v) = \delta(s, v)$
 quando $v \in Sol$.

$d(s) = \delta(s, s) = \emptyset$ base inductive

IF: primo modo per cui $d(\bar{v}) \neq \delta(s, \bar{v})$

Sol Dijkstra, $d(\bar{v})$



$d(w) \geq d(\bar{v})$ \downarrow minimo
 perché estratto \downarrow dalla coda

\rightarrow cammino minimo
 estratto \rightarrow fine $d(v)$

quando estratto $v \in Q \Rightarrow \forall \bar{v}$

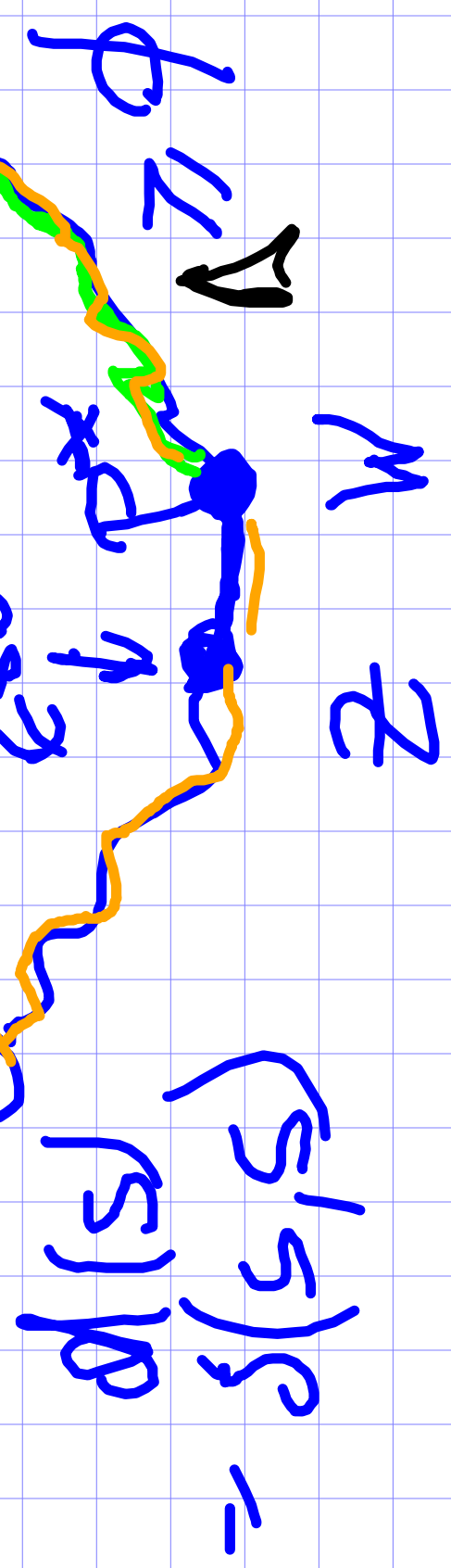
ultimo modo $\in Sol$ (\exists sempre perché $\in Sol$)

The above
 prove the
 correctness

$$\bar{v} \leftarrow \text{extract-min}$$

$$d(\bar{v}) \leq d(w)$$

$$\text{can Dist to } \bar{v} \quad \bar{v} \quad d(\bar{v}) \neq s(s, \bar{v})$$

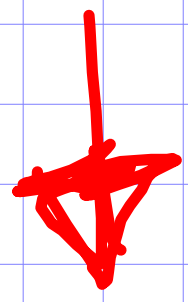


$$d(w) = d(z) + d(w, z)$$

$$\text{cost}(p) = d(w) + \Delta \geq d(\bar{v})$$

Dijkstra's algorithm
 no negative edges
 pre-processed

$$d(v) = 2$$



stopped!

because negative
 edges

$$d(w) = 4$$

$$d(u) + \Delta \geq d(v)$$

✓

OBS

$\pi \rightsquigarrow \mu \rightarrow \nu$

$$S \quad \boxed{(\mu, \nu)} \in \delta(s, \nu)$$

$$\cdot \rightsquigarrow \mu \quad \cdot \xrightarrow{\quad} \nu$$

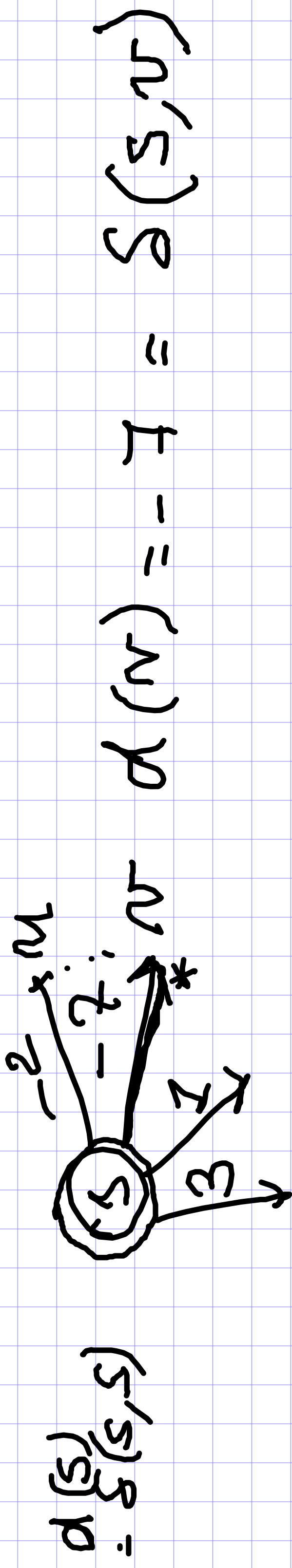
\cdot

S

quando si aggiunge μ e si trova (μ, ν)
allora nella coda ha il valore attimo
 $\delta(s, \nu)$

Dijkstra può tollerare archi di
costo negativo essenti alla
lunghezza

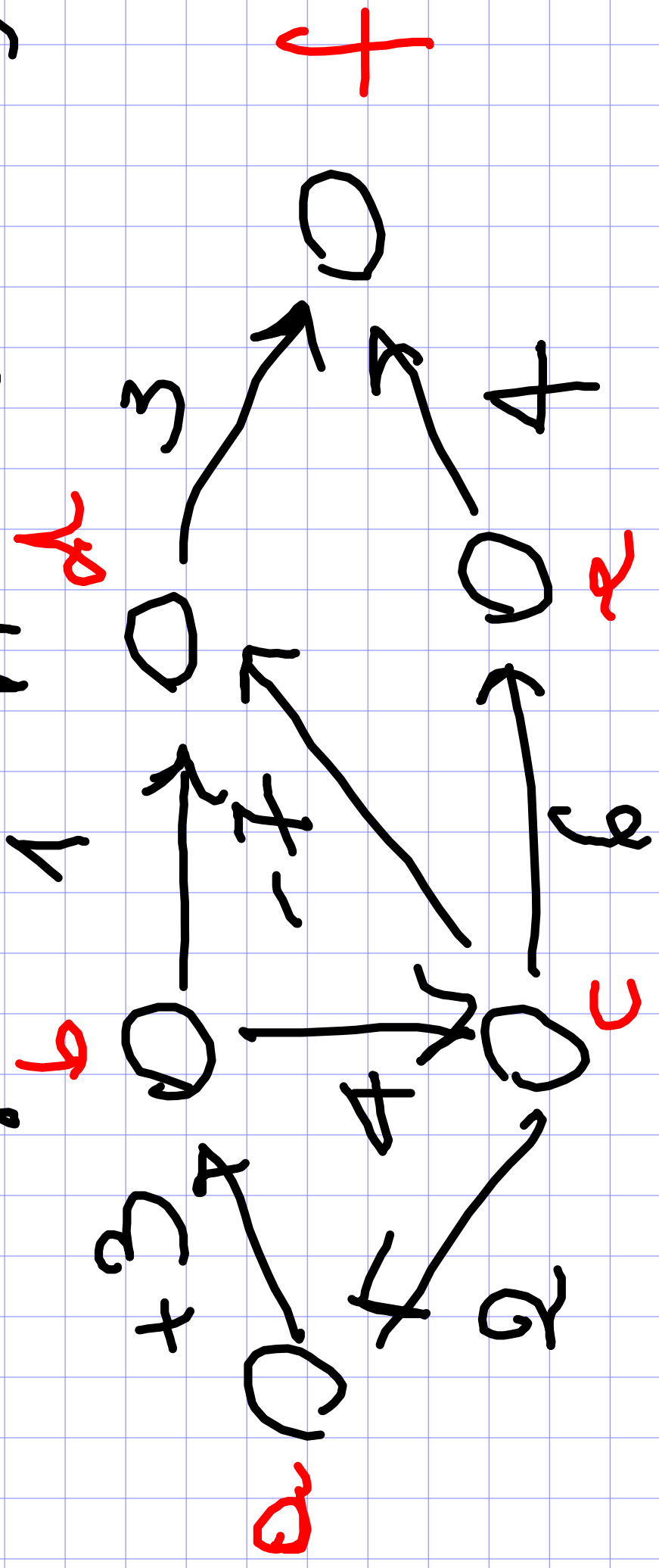
$$w: E \rightarrow \mathbb{R}^+$$



$$d(s) = \delta(s, s)$$

È ricorrenza per cammino minimo
che usa arco essenti da s a v
 $d(v) = \delta(s, v)$.

Example in our case application Dijkstra



$$G = (V, E)$$

$$w: E \rightarrow R$$

See earlier
 $w(e, f) = -4$
 now for the
 main Affixes
 Dijkstra

Dijkstra

see

Not

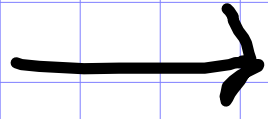
Cammini minimi $w: E \rightarrow \mathbb{R}$
(Tutte anche con costo negativo)
 \Rightarrow rischio di avere cicli di
costo negativo

Belman ford $(G, w: E \rightarrow R):$ boolean
TRUE se \exists ciclo di costo negativo
FALSE se \nexists ciclo di costo negativo
 $S(V, w) \forall v \in V$

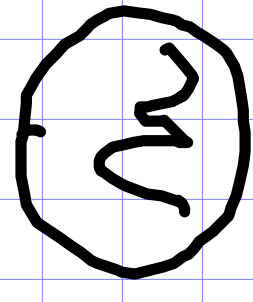
(#ordi)
OBS lunghezza cammino
minimo $\geq n$
 $\Rightarrow \exists$ ciclo costo negativo

It's a path

$$\# \text{ vertices} = a - b - c - d + 1$$



$$\# \text{ vertices} = n - 1 + 1$$



$$\# \text{ vertices} =$$

$$+ 1$$

→ 2 iterations
→ 2 vertices

End

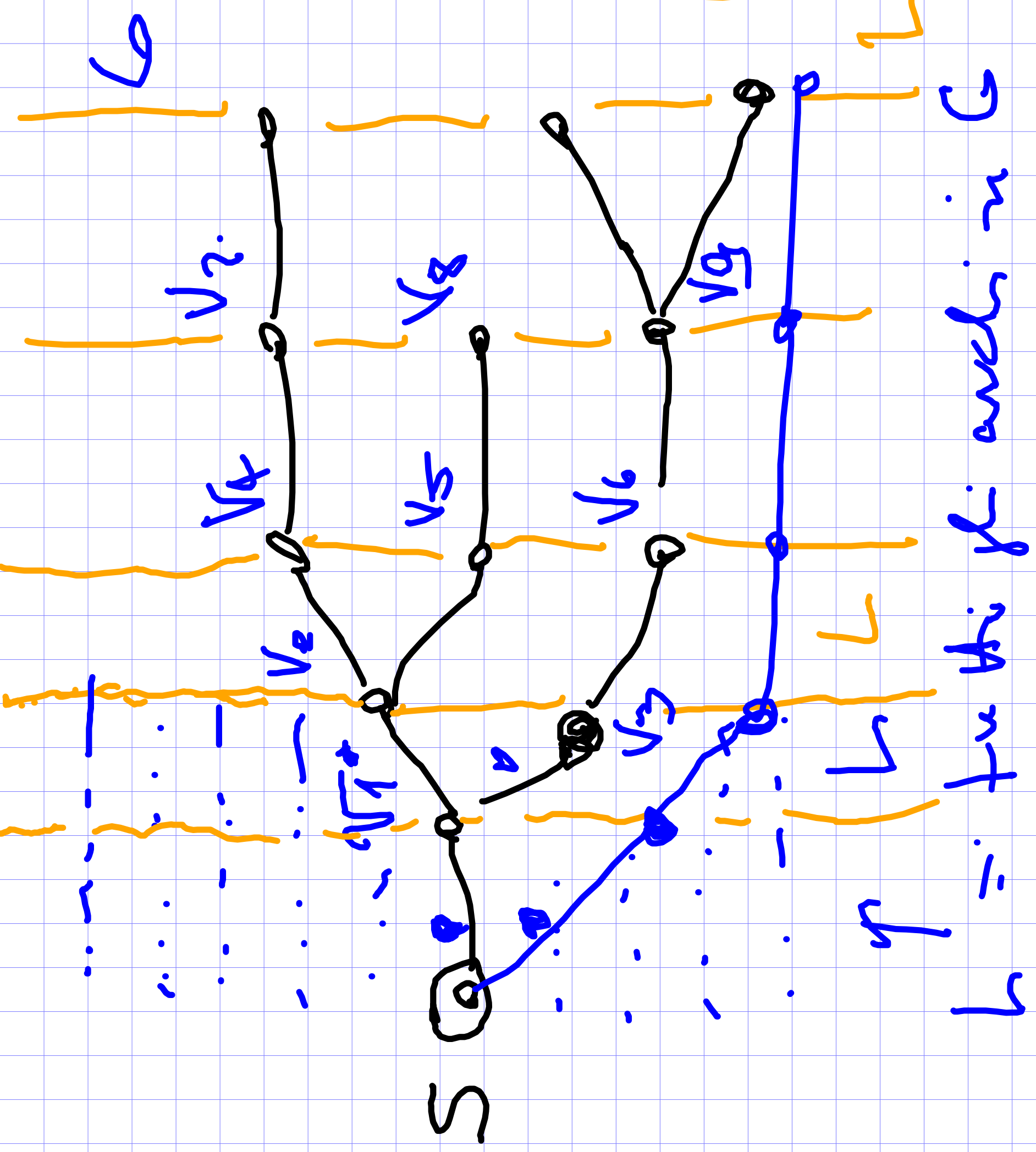


Bellman-Ford ($G = (V, E)$; $w: V \rightarrow \mathbb{R}$): boolean
INITIALISE list L di tutti gli archi in L .
{ Crea una lista L di tutti gli archi

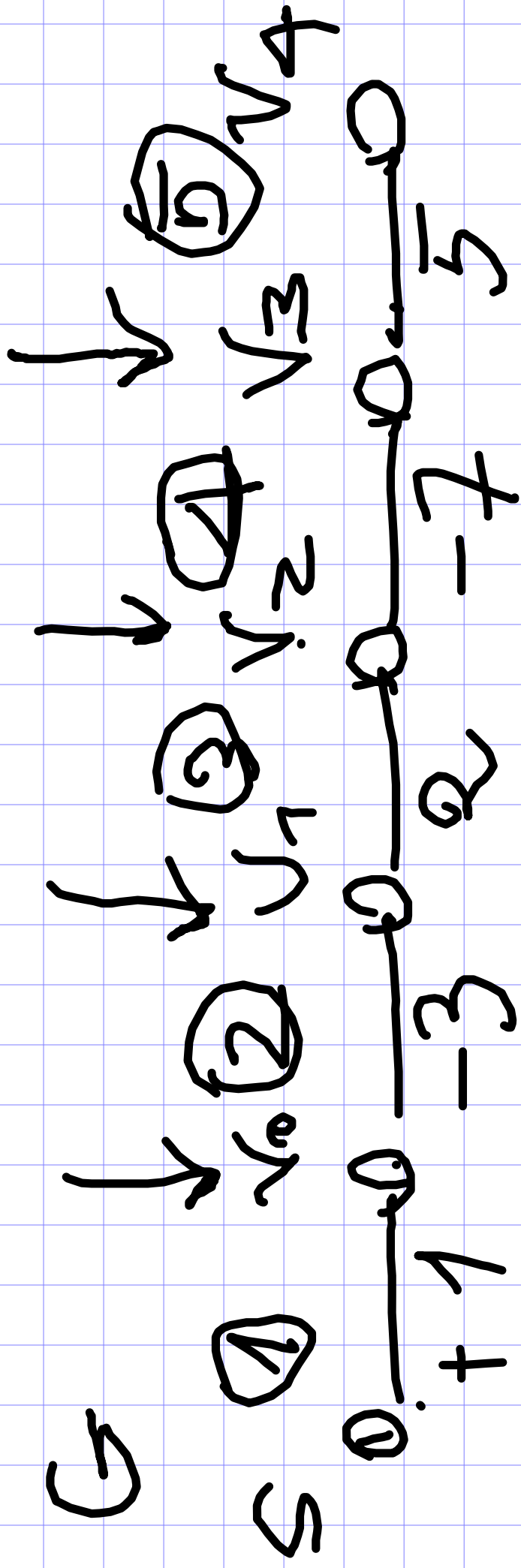
1°
for $i \leftarrow 1$ to $|V| - 1$
do
 rilasciano tutti gli archi in L
 nella ordine per distanza
 for $e = (u, v) \in L$ do
 if $d(v) > d(u) + w(u, v)$ then
 return TRUE
return FALSE
}

Albero
dei commi
minimi

Illustrazione
per provare
di correttezza



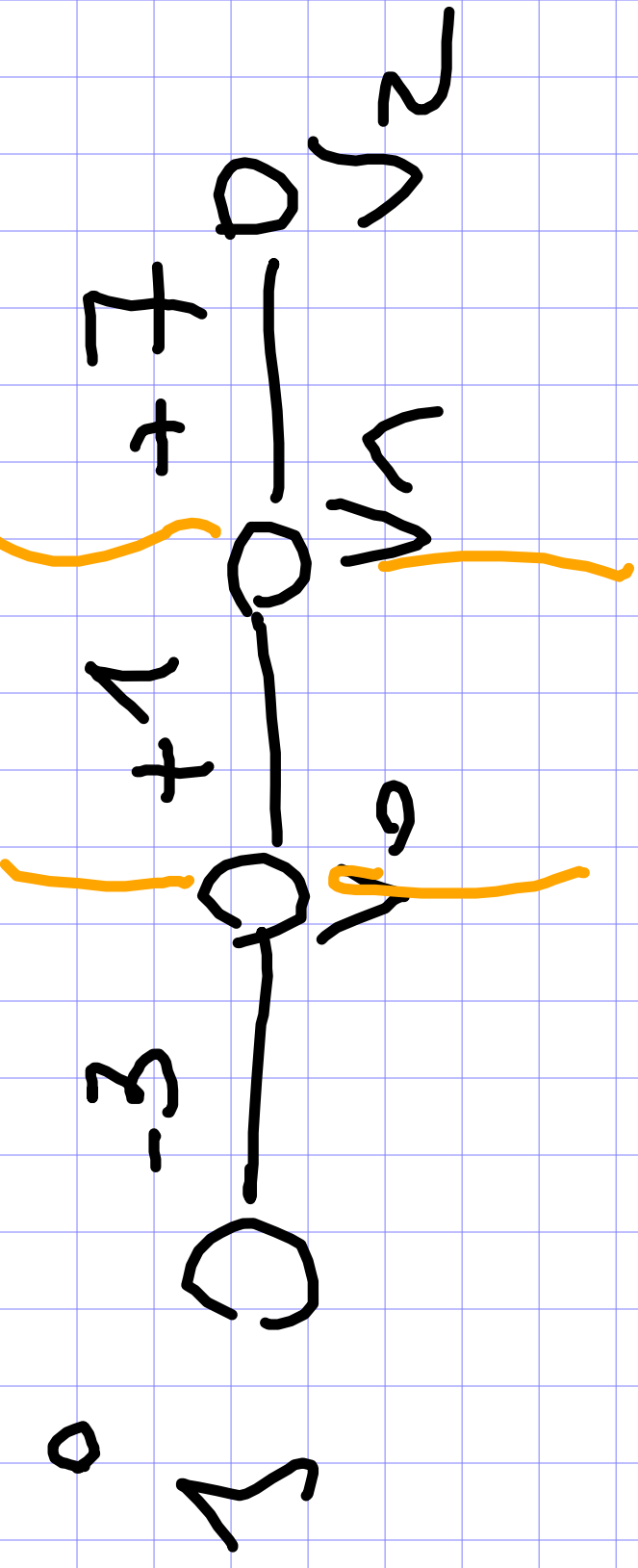
Ex 1.0



$$|V| = 5$$

$$L = \{(\underline{u, v_0}); (\underline{v_0, v_1}); (\underline{v_1, v_2}); (\underline{v_2, v_3}); (\underline{v_3, v_4}); (\underline{v_4, v_5})\}$$

	v_0	v_1	v_2	v_3	v_4
d	0	-2	0	-7	2
π		v_0	v_1	v_2	v_3



$$L = \left\{ \begin{pmatrix} v_1, v_2 \end{pmatrix}; \begin{pmatrix} v_0, v_1 \end{pmatrix}; \begin{pmatrix} s, v_0 \end{pmatrix} \right\}$$

	v_0	v_1	v_2
d	0	-3	-5
π	0	s	v_1

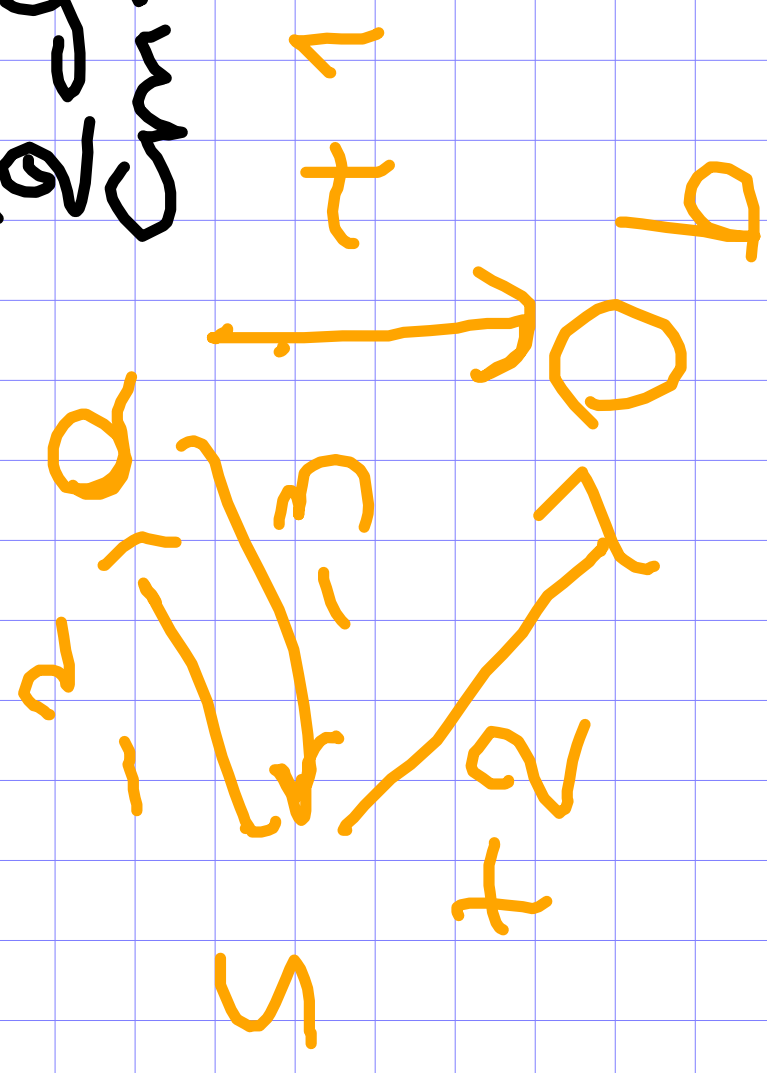
IterATA

1 2 3 4

Esempio $G = \text{petra}$
 L è l'elenco dei sensori
 4 istanze per sensori
 cui sono cicli: v_0, v_1, v_2 e s

~~4~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ ~~10~~ ~~11~~ ~~12~~ ~~13~~ ~~14~~ ~~15~~ ~~16~~ ~~17~~ ~~18~~ ~~19~~ ~~20~~ ~~21~~ ~~22~~ ~~23~~ ~~24~~ ~~25~~ ~~26~~ ~~27~~ ~~28~~ ~~29~~ ~~30~~ ~~31~~ ~~32~~ ~~33~~ ~~34~~ ~~35~~ ~~36~~ ~~37~~ ~~38~~ ~~39~~ ~~40~~ ~~41~~ ~~42~~ ~~43~~ ~~44~~ ~~45~~ ~~46~~ ~~47~~ ~~48~~ ~~49~~ ~~50~~ ~~51~~ ~~52~~ ~~53~~ ~~54~~ ~~55~~ ~~56~~ ~~57~~ ~~58~~ ~~59~~ ~~60~~ ~~61~~ ~~62~~ ~~63~~ ~~64~~ ~~65~~ ~~66~~ ~~67~~ ~~68~~ ~~69~~ ~~70~~ ~~71~~ ~~72~~ ~~73~~ ~~74~~ ~~75~~ ~~76~~ ~~77~~ ~~78~~ ~~79~~ ~~80~~ ~~81~~ ~~82~~ ~~83~~ ~~84~~ ~~85~~ ~~86~~ ~~87~~ ~~88~~ ~~89~~ ~~90~~ ~~91~~ ~~92~~ ~~93~~ ~~94~~ ~~95~~ ~~96~~ ~~97~~ ~~98~~ ~~99~~

$2 \rightarrow a$
 $5 \rightarrow 3$
 Escaping cycle



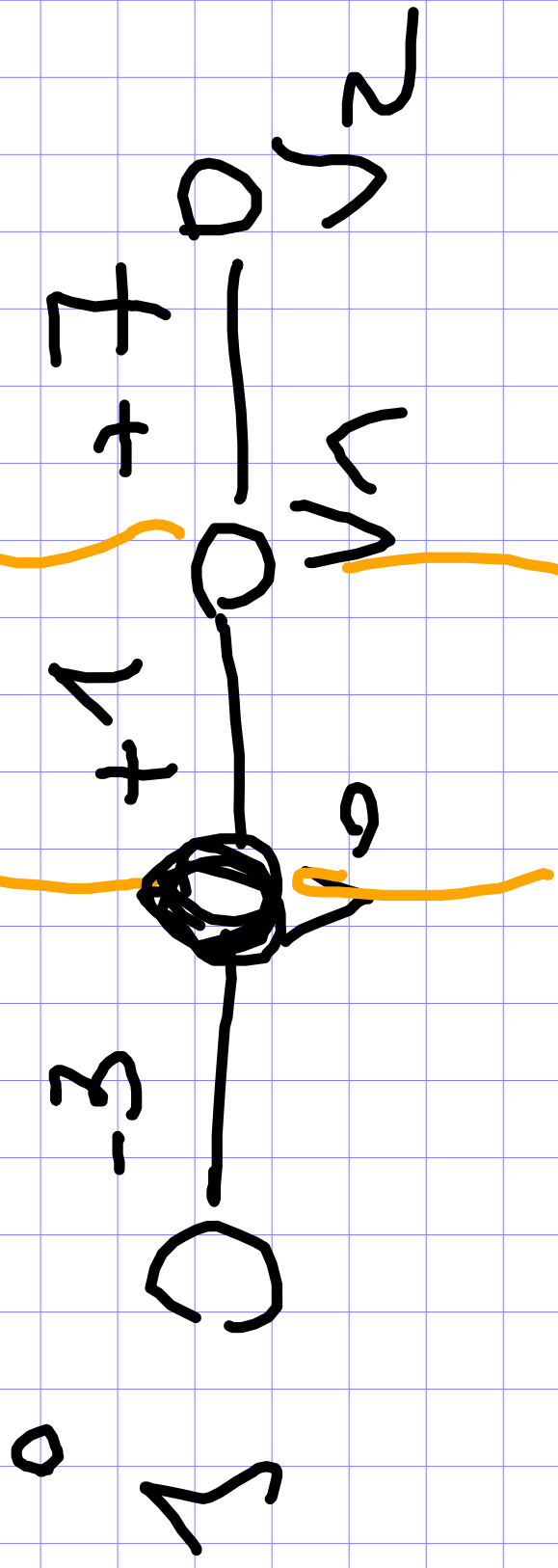
Every step takes
 like
 cycle
 all the 4th
 if needed

a		b	
2	5	2	5

s		a	
2	5	2	5

1 2
 Cycle!

s		a	
2	5	2	5



$$\pi$$

S	v_0	v_1	v_2
1	-3	-2	$+5$
\emptyset	S	v_0	v_1

Grado è per poter

Esempio in cui con una sola

iterazione si trova l'albero dei componenti

$$L = \left\{ \underbrace{(5, v_0)}, \underbrace{(v_0, v_1)}, \underbrace{(v_1, v_2)} \right\}$$

ITERATA

1

2

Correttura

Se comosco che sovrano ottimo ab $S \rightarrow u$
è esultante so q_i anche

$(1, v) = e_1, e_2, \dots, e_n = (z, u)$
e rilanza nella 1. ordine

e_1, e_2, \dots, e_r

trovo il costo $\alpha(u) = \delta(s, u)$

Non conosco l'ordine per cui fu
aperta l'azione e provo tutti gli

avoli e ricorro mente perché fu
l'arco che è l'ultimo del
cammino ottimo.

$$n = |V|$$

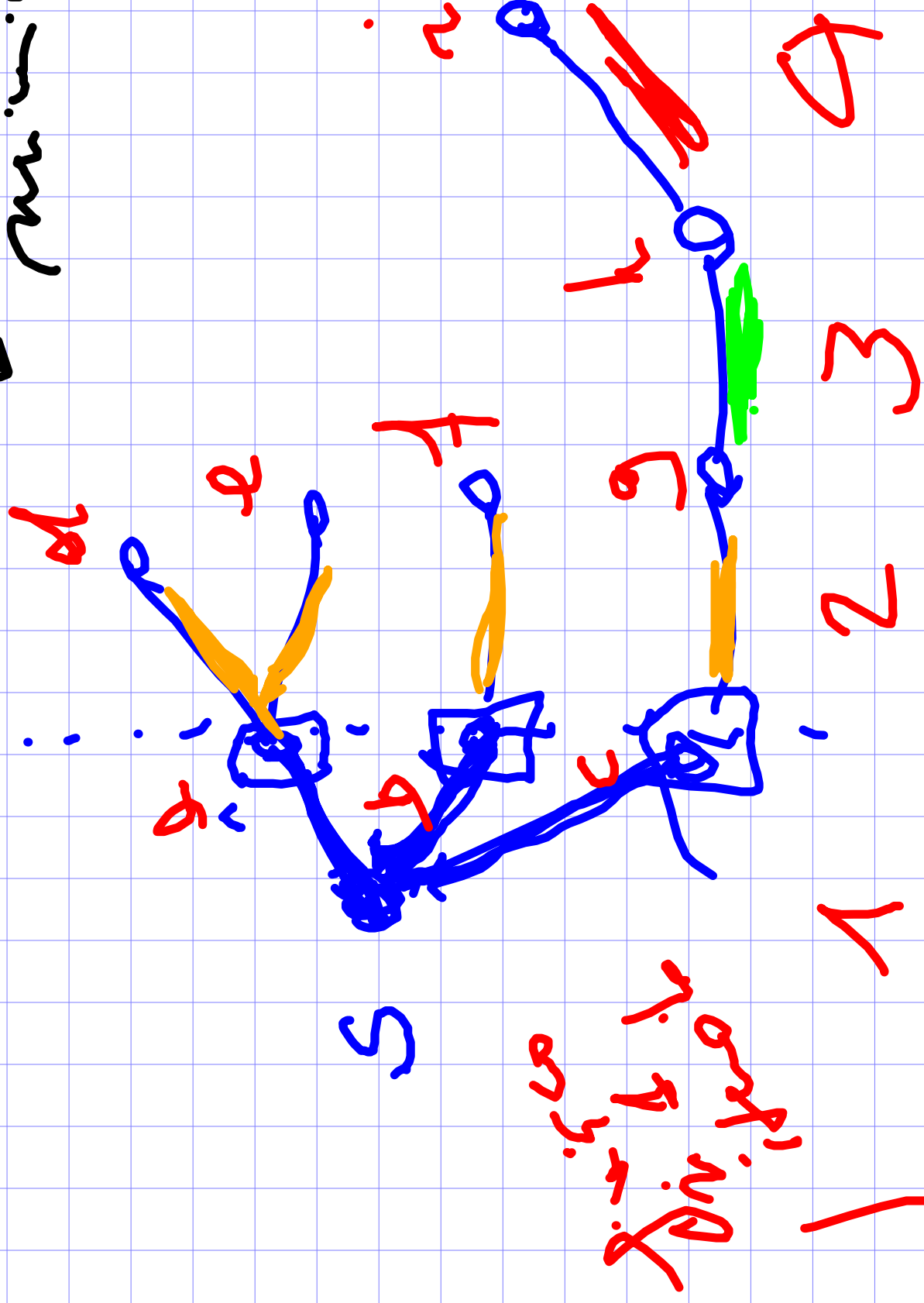
Caminhi minimi sono semplici.
Hanno $\leq n-1$ archi.

Se trovo che zilero in arco
in posizione n (ossia trovo
un cammino minimo di lunghezza
 n) allora $\exists (TRUE)$ un cammino
con un ciclo di costo negativo.

\Rightarrow soluzione del problema

Calculus
 differentiation
 minimization

\rightarrow
 step by step minimization
 system to
 find a distance
 ≤ 1



$$L = (s, a)(s, b)(s, c)(s, d)(s, e)(s, f)(s, g)(s, h)(s, i)(s, j)(s, k)(s, l)$$

Confronti in tempo

Ciclo lista $|E|$
visiamo tutti gli archi $|V| - 1$ volte

Almeno ancora gli archi
 $O(V E)$ $O(E)$

Confronti $O(V E)$
Non differisce dalle rappresentazioni.

examini minimi de
originate singula

[examini minimi
if copy a di verten
if $v \in V$ start job
non concept

[Floyd Wars hell
METRO TRAFFIC

[Algorithmi di Johnson

