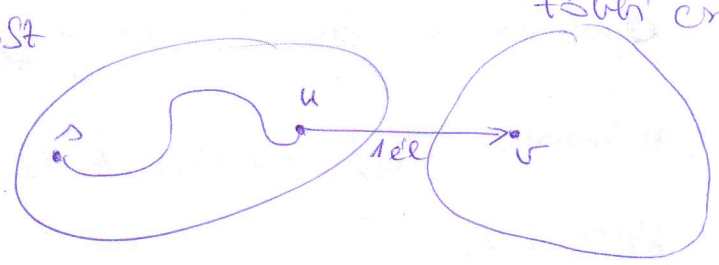
2018. 12. 06. G.

DIJKSTRA ALGORITHMUS:

Input: G irányított gráf
$c(f) \geq 0$ élsúlyokkal

s forráscsúcstól minimális út meghatározása $\forall$ más csúcsba

elve:



távolság $[u]$ = s-ből u-ba menő minimális út hossza

$d[v]$ = min olyan út hossza $s \rightsquigarrow v$-be
ahol $s \rightsquigarrow u$ min és végig KÉSZ-ben van
és utána 1 éllel $u \to v$ KÉSZ

honnan $[v]$ = melyik KÉSZ-beli csúcsból jön a legrövidebb út

pótZH lehet

vizsgát előtti nap mindig tanulási idő

lehet, hogy lesz mintavizsga

pl.: elméleti tétel
- algoritmus futtatása
- algoritmus leírása

előző ZH-n kívüli anyag is

KÉSZ = { s }

távolság [u] = { 0 ha v = s ; ∞ * egyébként }

$$d[v] = \begin{cases} * & \text{ha } v = s \\ c(s,v) & \text{ha } s \to v \text{ él van} \\ \infty & \text{különben} \end{cases}$$

honnan [v] = { s ha v = s ; s ha s → v él létezik ; * különben }

while van olyan csúcs amire $d[\,] \neq *, \infty$

v^* := minimális $d[\,]$ értékhez tartozó

v^* : KÉSZ-be

távolság $[v^*]$:= $d[v^*]$

$d[v^*] = *$

(*)
for w in v^* szomszédai:
  if $d[w] \neq *$: # $w \notin$ KÉSZ
    if távolság $[v^*] + c(v^*, w) < d[w]$:
      $d[w]$ = távolság $[v^*] + c(v^*, w)$
      honnan $[w] = v^*$

LSz: ha G mátrixosan adott:

$O(n^2)$

- eleje: $O(n)$
- While ciklus: $\leq n-1$ ciklus van (mert minden körben 1 csúcs kerül kitörlésre)
- 1 ciklusban: 1 min keresés: $O(n)$
  - távolságjavítás: $O(1)$
  - (*): $5 \cdot n$ mert v^* minden szomszédját végigjárjuk a mátrixban
  - $\rightarrow O(n)$

ha G éllistával adott:

eleje $O(n)$

While ciklusok összesen:

$(n-1)$ minimumkeresés $O(n^2)$

de (*) v^* esetén $O(d(v^*))$ lépés $\Rightarrow O(e)$ összes él

---

$O(n^2+e) = O(n^2)$

ha $d[v]$ nem tömb hanem heap/lupac akkor a minimumkeresés $O(n)$ helyett $O(\log n)$

$\Rightarrow$ teljes algoritmus $O(e \cdot \log(n))$ lesz

Helyesség: kell: távolság[v] minden v-re jó,

azaz távolság[v] = minimuma a lehetséges $s \rightsquigarrow v$ utak hosszának

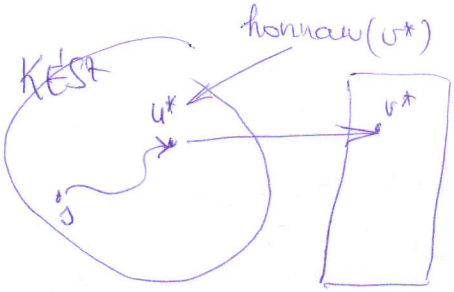
bizonyítás: indukcióval a KÉSZ-be kerülés sorrendjére

1.) távolság[s] = 0 helyes mert nincs negatív él

2.) Tfh.: $\forall$ távolság[u] helyes egy adott pillanatban $\Rightarrow$ a következő csúcsra amit KÉSZ-be kerül (v^*) erre is igaz

távolság[v^*] = min $s \rightsquigarrow v^*$ út hossza

$d[v^*]$ = távolság[u^*] + $c(u^*, v^*)$



Kell (A) van távolság

távolság[u^*] + $c(u^*, v^*)$ hosszúságú út $s \rightsquigarrow v^*$-ba

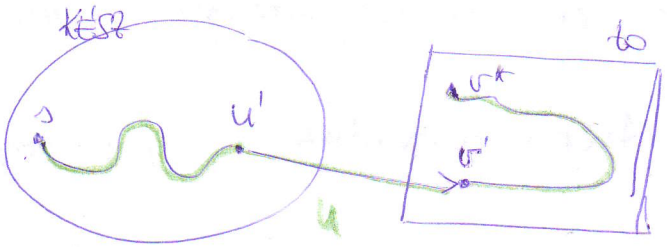
(B) $\forall$ út $s \rightsquigarrow v^*$-ba

$\geq d[v^*]$ = távolság[u^*] + $c(u^*, v^*)$

(A): v^*-ba út, ami minimális:

távolság[u^*] + $c(u^*, v^*)$

(B):



u hossza = $s \rightsquigarrow u'$ út hossza + $c(u',v')$ + $v' \rightsquigarrow v^*$ hossza $\geq$

① $\geq$ távolság[u'] + $c(u',v')$ + $v' \rightsquigarrow v^*$ hossza $\geq$

② $\geq$ d[v'] + $v' \rightsquigarrow v^*$ hossza $\geq$ d[v'] $\geq$ d[v*] ③

① mert távolság[u'] helyes ⇒ ∀ út u'-be $\geq$ távolság[u'] hosszú

② mert d[v'] = min olyan út hossza ami KÉSZ-ből 1 éllel megy v'-be

③ mert d[v*] minimális nem d[v']

Feladat:
- adott n darab város, köztük közvetlen utak
- útfelújítást tervezni minden közvetlen útra tudjuk a költséget
- cél: bárhonnan bárhova el lehessen jutni felújított úton (de nem feltétlenül közvetlen út) legkevesebb költséggel

feladat általánosítása:
input: adott [redacted] n csúcsú irányítatlan gráf összefüggő [redacted] (nemnegatív élekkel)

minimális összsúlyú részt (néhány élet) amik

feszítőfa
- összefüggő részt alkotnak
- minden csúcsot lefednek
- körmentes → ha lenne kör akkor egy élét el lehetne hagyni

Algoritmus: (mohó módszer): Prim algoritmus

- elindulni s csúcsból innen kiinduló fa építése ~~a következő módon~~ addig amíg el nem érjük az összes csúcsot

Pontosabban: - LEFEDVE halmaz azokat már elérte már a fa
- minden lépésben a min élet ami kilép LEFEDVE halmazból azt bevenni az épülő fába

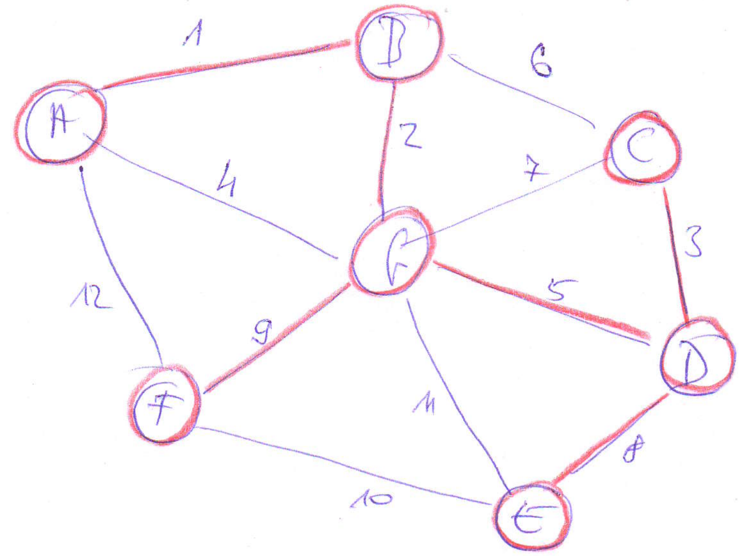
LEFEDVE = {s}

F = ∅ # épülő fa

---

while van LEFEDVE-ből kilépő él:
  min súlyú él (u,v) LEFEDVE
  (u,v) F-be
  v LEFEDVE-be

példa.



F	LEFEDVE
–	A
AB	B
BG	G
GD	D
DC	C
DE	E
GF	F

Be lehet bizonyítani, hogy [KS7]: $O(n^2)$ (mátrix) / $O(e \cdot \log n)$ (éllista)

hasonlóan a Dijkstra-hoz

Helyesség: . . .

↑ VIZSGAANYAG IDÁIG

További tananyag: – Rónyai – Szabó – Ivanyos tankönyv

= Coursera online tananyag

pl.: Stanford: Algorithms

4–8 perces videók

ellenőrző kérdésekkel

házifeladatokkal programozási feladatok

(bizonyítás nélkül, ingyenes)