Brzi algoritmi za problem najkraćeg puta

Andrija Mandić

Prirodoslovno-matematički fakultet — Matematički odsjek

22. rujna 2021.

Uvod

Problem najkraćeg puta široko primjenjiv u stvarnom svijetu:

- navigacijski sustavi, prometnice,
- *web* pretraživanje,
- usmjeravanja podataka na internetu,
- baze podataka.

Problem dijelimo u dvije faze:

– faza *predprocesiranja*,

– faza *upita*.

Sadržaj

Osnovni pojmovi teorije grafova

Definicija

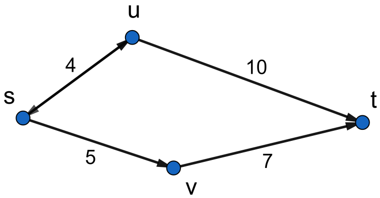
Usmjereni težinski graf G je uređena trojka $G = (V, E, \omega)$, gdje je V skup vrhova, $E \subseteq V \times V$ skup bridova grafa G, te $\omega\colon E \to R$ funkcija koja svakom bridu pridružuje njegovu težinu.

Definicija

Put $P = \langle v_0, v_1, v_2, \ldots, v_k \rangle$ od vrha v_0 do vrha v_k je niz vrhova redom povezanih bridovima. Težina puta iznosi:

$$\omega(P) = \sum_{i=0}^{i=k-1} \omega(v_i, v_{i+1}).$$

Osnovni pojmovi teorije grafova



Slika: Primjer usmjerenog grafa

Definicija

Najkraći put, tj. minimalnu težinsku udaljenost između vrhova $u, v \in V$ označavamo s $d_G(u, v)$ te ona iznosi:

$$d_G(u,v) := \begin{cases} \min\{\,\omega(P) \mid P \text{ je put od } u \text{ do vrha } v\,\} & \text{ako} \\ \qquad\text{postoji put } P \text{ od } u \text{ do vrha } v & \\ +\infty & \text{inače} \end{cases}.$$

Varijante problema

Vrijednosti koje mogu poprimiti težine bridova? Smiju li biti negativne? Kolika je očekivana gustoća grafa? Tražimo li aproksimativno ili egzaktno rješenje?

Problemi najkraćeg puta u grafu:

- **Najkraći put iz jednog vrha** (eng. *Single-Source Shortest-Path Problem*)
- **Najkraći put do jednog odredišta** (eng. *Single-Destination Shortest-Path Problem*)
- **Najkraći put između dva vrha** (eng. *Single-Pair Shortest-Path problem*)
- **Najkraći put između svaka dva vrha** (eng. *All-Pairs Shortest-Paths Problem*)

Dijkstrin algoritam – ideja

- Vrsta grafa: težinski graf s nenegativnim težinama bridova.
- Algoritam kreće iz početnog vrha te u svakoj iteraciji pronađe najkraći put do najbližeg do tad nepronađenog vrha.
- Skup svih vrhova idejno se dijeli na tri dijela: *neposjećene*, *posjećene* i *obrađene* vrhove.
- Iterativno se obnavlja polje D veličine $|V|$, te za svaki vrh v vrijedi $D[v] \geq d_G(s, v)$, gdje je s početni vrh. Jednakost vrijedi ukoliko je v *obrađen*.

Dijkstrin algoritam – ideja

1. Pronađi vrh v najbliži početnom, a da u njemu nisi bio (*posjećen* ali *neobrađen*).
2. Zabilježi da si bio u vrhu v – *obrada*.
3. Pokušaj do svih vrhova ići preko vrha v i popraviti rješenje.

Ponavljaj dok postoji takav vrh v.

Pretraživanje A^*

- Koristi heurističku funkciju $\pi_t \colon V \to \mathrm{R}$ kako bi usmjerio pretraživanje prema ciljnom vrhu t.
- Algoritam sličan kao kod Dijkstrinog pretraživanja. Razlika je odabir novog vrha u iteraciji s najmanjom vrijednosti procjene ukupne duljine puta: $k(v) = D[v] + \pi_t(v)$.

Definicija

Neka su zadani graf $G = (V, E, \omega)$ i funkcija $\pi \colon V \to \mathrm{R}$. *Reducirana funkcija težine bridova* s obzirom na funkciju π definira se kao:

$$\omega_\pi(v, w) = \omega(v, w) - \pi(v) + \pi(w).$$

Lema

Neka je zadan graf $G = (V, E, \omega)$, funkcija $\pi \colon V \to \mathrm{R}$ i pripadna reducirana funkcija težine bridova ω_π. Za proizvoljan put $P = \langle v, x_1, \ldots, x_k, w \rangle$ vrijedi:

$$\omega_\pi(P) = \omega(P) + \pi(w) - \pi(v).$$

Definicija

Funkciju $\pi \colon V \to \mathrm{R}$ nazivamo *izvedivom* ako je njena reducirana funkcija težine nenegativna za svaki brid grafa G, tj. $\omega_\pi(v, w) \geq 0$ za svaki $(v, w) \in E$.

Primjer izvedive heurističke funkcije: euklidska udaljenost.

Lema

Neka je funkcija $\pi \colon V \to \mathrm{R}$ izvediva takva da za neki vrh $t \in V$ vrijedi $\pi(t) \leq 0$. Tada za svaki vrh $v \in V$ vrijedi nejednakost $\pi(v) \leq d_G(v, t)$.

Pretraživanje A^*

- Pretraživanje koje odabir novog vrha temelji na najmanjoj vrijednosti $k(v) = D[v] + \pi_t(v)$ ekvivalentno je Dijkstrinom algoritmu s reduciranom funkcijom težine ω_{π_t}.
- Težina brida (u, v) zamjenjuje se s $\omega(u, v) - \pi_t(u) + \pi_t(v)$.
- $d_\omega(v, t) = d_{\omega_{\pi_t}}(v, t) + \pi_t(v) - \pi_t(t)$.
- Korektnost A^* algoritma: izvedivost funkcije π_t, tj. $\omega_\pi(u, v) \geq 0$, za svaki $(u, v) \in E$ + Dijkstin algoritam.

Dvosmjerno pretraživanje

- Pretraga *unaprijed*.
- Pretraga *unazad*.
- Alterniranje koraka pretraživanja ili paralelno izvođenje odabirom najprikladnijeg vrha.

Uvjeti zaustavljanja

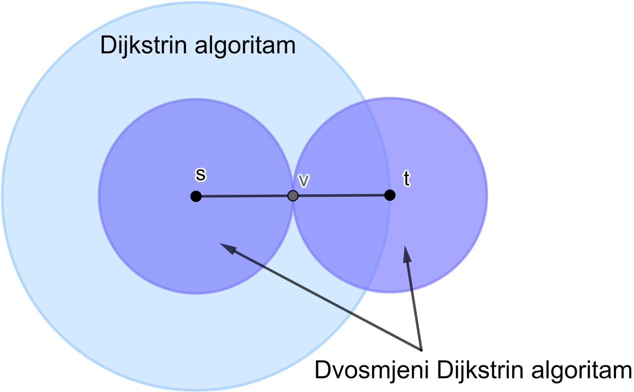
Dijkstrina dvosmjerna pretraga:

- zaustavlja se kada je neki vrh v obrađen u pretrazi unaprijed i unazad.
- duljina nakraćeg puta iznosi $\min\{D_s[w] + D_t[w] \colon w \text{ posjećen u obje pretrage}\}$.

Dvosmjerno pretraživanje A^*:

- *konzistentne* heuristike $\pi_t + \pi_s = const.$ – zaustavljenje nakon obrađivanja istog vrha u obje pretrage.
- *simetrični pristup* – ukoliko jedno od pretraživanja obradi vrh v za koji $D_s[v] + \pi_t(v) \geq \mu$ ili kada nema posjećenih vrhova za obradu. Vrijednost μ je duljina trenutnog pronađenog rješenja.

Ilustracija prostora pretrage



Slika: Usporedba prostora pretrage klasičnog i dvosmjernog Dijkstrinog pretraživanja.

Contraction Hierarchies – ideja

Hijerarhijski pristup problemu.

Predprocesiranje:

- računanje hijerarhije vrhova, funkcija $rank \colon V \to \mathbb{N}$.
- dodavanje novih potrebnih bridova – *prečaca*.

Faza upita:

- dvosmjerno Dijkstrino pretraživanje.
- obje pretrage proširuju se na vrhove uzlazno po *rank* vrijednosti.

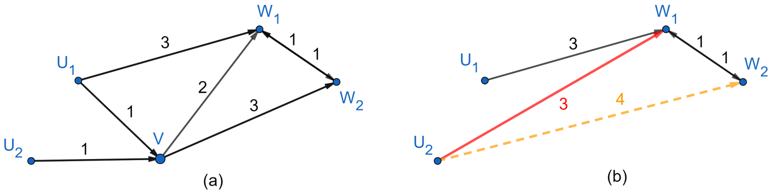
Kontrakcija vrha

- Osnovni postupak faze predprocesiranja.
- Izbacuje se vrh i njemu incidentni bridovi iz trenutnog grafa. Dodaju se svi potrebni bridovi kako bi se sačuvali najkraći putovi u preostalom grafu.

Izbacivanje vrha v:

- neka su $\{u_1, u_2, \ldots, u_l\}$ ulazni vrhovi, $(u_i, v) \in E$,
- neka su $\{w_1, w_2 \ldots, w_k\}$ izlazni vrhovi, $(v, w_j) \in E$,
- za svaki par u, w dodajemo brid (u, w) težine $\omega(u, v) + \omega(v, w)$ ukoliko je put $\langle u, v, w \rangle$ jedini najkraći put od u do v,
- izbacimo vrh v i njegove incidentne bridove.

Kontrakcija vrha – primjer



Slika: Primjer grafa prije (a) i poslije (b) kontrakcije vrha v.

Raspored vrhova

– Potrebno provesti kontrakciju nad svim vrhovima.
– Trenutni grafovi G_i, rezultantni graf G^*.
– Funkcija *rank* odgovara poretku kontrakcije vrhova.

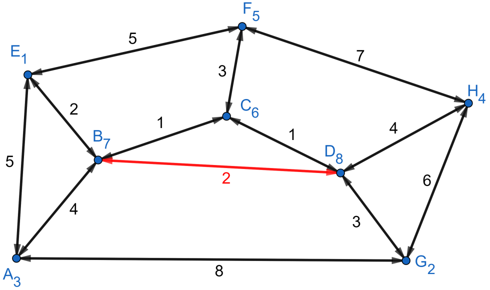
Određivanje poretka temelji se na:

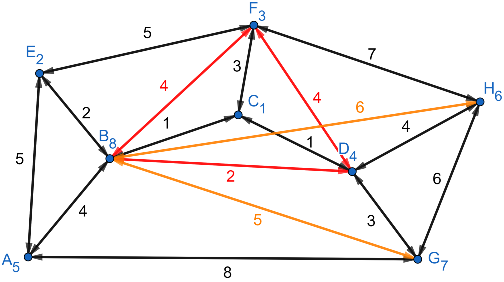
- *razlici bridova* (eng. *edge difference*),
- *uniformnosti* (eng. *uniformity*).

Obnavljanje prioriteta vrhova:

- *lijeno ažuriranje* (eng. *lazy update*),
- *ažuriranje susjeda* (eng. *neighbors recomputing*),
- *periodično ažuriranje* (eng. *periodically rebuilding*).

Utjecaj redoslijeda kontrakcija





Faza upita CH

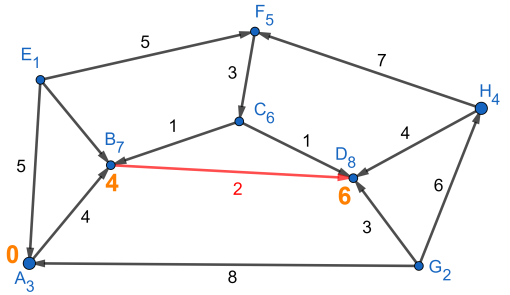
Definiramo:

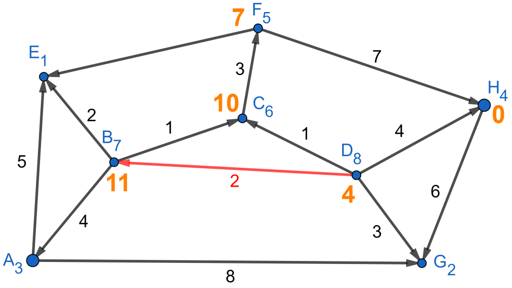
- *rastući podgraf* $G^*_\uparrow = (V, \{(u,v) \in E^* : rank(u) < rank(v)\}, \omega^*_\uparrow)$,
- *padajući podgraf* $G^*_\downarrow = (V, \{(u,v) \in E^* : rank(u) > rank(v)\}, \omega^*_\downarrow)$.

Pronalazak najkraćeg puta od vrha s do vrha t provodi se:

1. Dijkstrinim pretraživanjem unaprijed u grafu $G^*_\uparrow$, počevši od vrha s – rezultat polje D_s,
2. Dijkstrinim pretraživanjem unazad u grafu $G^*_\downarrow$, s početkom u vrhu t – rezultat polje D_t,
3. računanjem $d_G(s,t) = \min\{D_s[v] + D_t[v] : v \in I\}$, gdje je I skup svih obrađenih vrhova u obje pretrage.
4. rekonstrukcijom najkraćeg puta (ukoliko je potrebno).

Faza upita CH – primjer





Landmark A^* – ideja

Ciljno orijentirani pristup problemu.

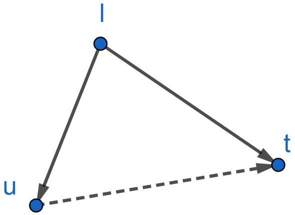
Definicija

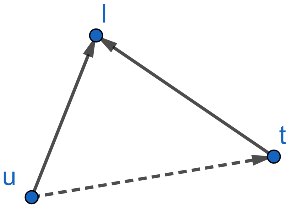
Vrh $v \in V$ nazvat ćemo *orijentirom* ukoliko su poznate vrijednosti najkraćih putova *do* i *od* svih preostalih vrhova u grafu G.

ALT algoritam:

- koristi poznate udaljenosti vezane za orijentire i ideju nejednakosti trokuta kako bi se dobila preciznija izvediva heuristička funkcija π_t,
- fazu upita realizira A^* pretraživanjem s heurističkom funkcijom dobivenom predprocesiranjem.

Modificirana nejednakost trokuta





$$d_G(l,t) \leq d_G(l,u) + d_G(u,t) \implies d_G(u,t) \geq d_G(l,t) - d_G(l,u)$$
$$d_G(u,l) \leq d_G(u,t) + d_G(t,l) \implies d_G(u,t) \geq d_G(u,l) - d_G(t,l)$$

$$\pi_{t,l}(u) := \max\big(d_G(l,t) - d_G(l,u),\ d_G(u,l) - d_G(t,l)\big) \leq d_G(u,t)$$

Odabir vrhova orijentacije

Načini odabira vrhova:

- *slučajni* odabir vrhova,
- *pohlepni najudaljeniji* odabir vrhova,
- *planarni* odabir vrhova.

Faza upita ALT

Neka je L odabrani skup vrhova orijentacije. Za proizvoljan vrh u, heurističku procjenu udaljenosti do ciljnog vrha t definiramo izrazom:

$$\begin{aligned}\pi_t(u) &:= \max_{l \in L} \pi_{t,l}(u) \\ &= \max_{l \in L} \{ \max(d_G(l,t) - d_G(l,u), d_G(u,l) - d_G(t,l)) \}.\end{aligned}$$

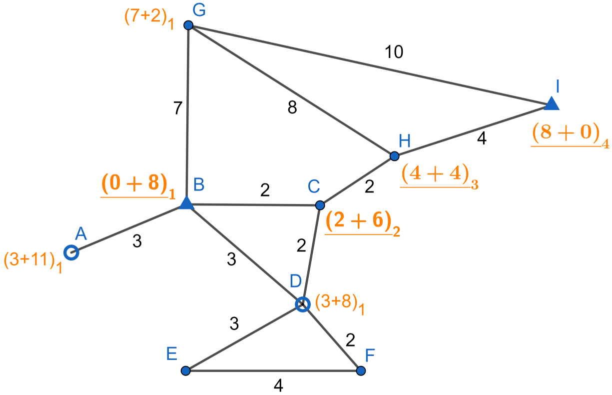
Lema

Za zadani skup vrhova orijentacije L, ciljni vrh $t \in V$, funkcija π_t je *izvediva*.

Faza upita provodi se A^* pretraživanjem s definiranom heurističkom funkcijom π_t.

Faza upita ALT – primjer

Pronalazak najkraćeg puta od vrha B do ciljnog vrha I. Skup orijentira $L = \{A, D\}$.



Faza upita ALT – primjer

Pronalazak najkraćeg puta od vrha B do ciljnog vrha I. Skup orijentira $L = \{A, D\}$.

$u \rightarrow$	A	B	C	D	E	F	G	H	I
$d_G(A,u)$, $d_G(u,A)$	0	3	5	6	9	8	10	7	11
$d_G(D,u)$, $d_G(u,D)$	6	3	2	0	3	2	10	4	8
$\pi_{I,A}(u)$	11	8	6	5	2	3	1	4	0
$\pi_{I,D}(u)$	2	5	6	8	5	6	2	4	0
$\pi_I(u)$	11	8	6	8	5	6	2	4	0
$d_G(u,I)$	11	8	6	8	11	10	10	4	0
$d_G(B,u)$	3	0	2	3	6	5	7	4	8

Tablica: Prikaz vrijednosti najkraćih putova i njihovih procjena, u upitu iz primjera.

Implementacija i testiranje algoritama

- Uspoređuju se Dijkstrin, *Contraction Hierarchies* i *Landmark* A^* algoritmi.
- Faza upita - pronalazak vrijednosti najkraćeg puta između 1000 slučajno odabranih parova vrhova.
- Skupovi podataka dobiveni su iz prometnih mreža.
- Pri implementaciji korišten je programski jezik *Java*.

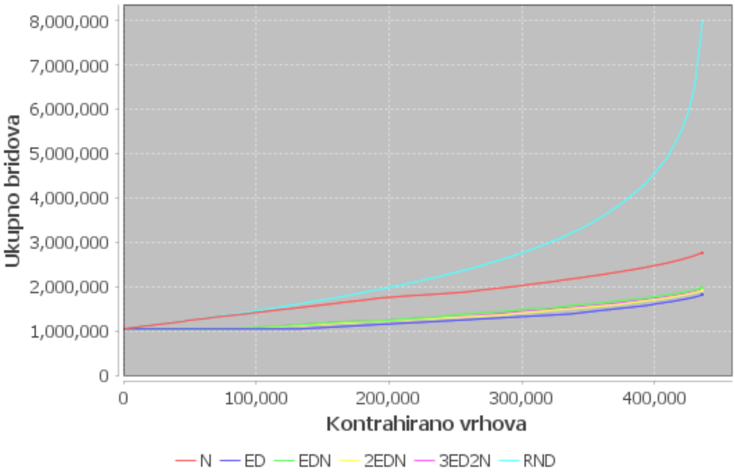
	Savezna država	Broj vrhova	Broj bridova
NY	New York	264 346	730 100
COL	Colorado	435 666	1 042 400
FLA	Florida	1 070 376	2 687 902
CAL	Kalifonija i Nevada	1 890 815	4 630 444

Tablica: Prikaz informacija o skupovima podataka cestovnih mreža.

Contraction Hierarchies

Trenutne grafove G_i te završni graf G realiziramo u jednoj strukturi, ignorirajući izbačene vrhove i bridove.

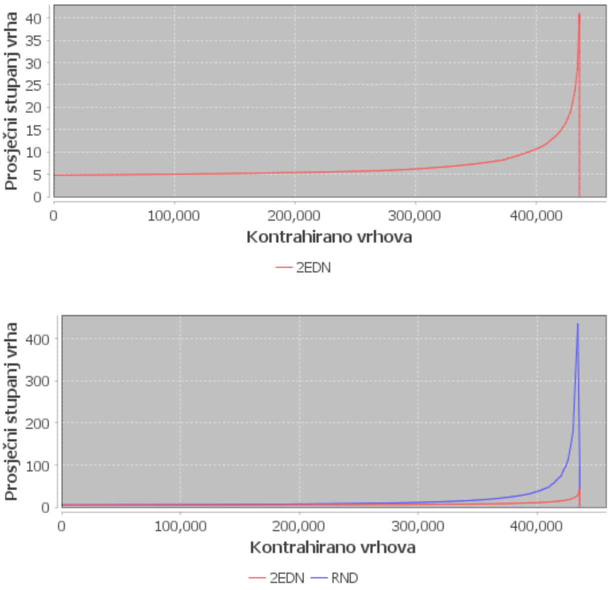
- Označimo za $v \in V$ s $|V_{in,v}|$ broj ulaznih, te s $|V_{out,v}|$ broj izlaznih vrhova.
- Procjena razlike bridova: $ED_v = |V_{in,v}| \cdot |V_{out,v}| - |V_{in,v}| - |V_{out,v}|$.
- Označimo s N_v broj obrisanih susjeda pojedinog vrha v.



Slika: Usporedba ukupnog broja bridova rezultantnog grafa G^* dobivenog različitim vrijednostima prioriteta vrhova. Korišten COL skup podataka.

	Prioritet	Dodano bridova	$t_{predproc}$	t_{pUpit}	$N_{pObrađeno}$
N	N	1 722 113	29 min 10.35 s	3.407 ms	246.11
ED	ED	767 267	14 min 31.47 s	3.716 ms	672.46
EDN	$ED + N$	929 317	16 min 36.30 s	3.430 ms	402.42
2EDN	$2ED + N$	866 272	16 min 5.85 s	3.429 ms	444.11
3ED2N	$3ED + 2N$	888 866	16 min 27.91 s	3.458 ms	459.94
RND	nasumično	6 921 552	12 h 35 min 2.06 s	58.991 ms	2761.80

Tablica: Prikaz informacija o predprocesiranju i fazi upita za različite načine određivanja prioriteta vrhova pri odabiru za kontrakciju. Korišten COL skup podataka.



Slika: Usporedba prosječnog stupnja vrha trenutnog grafa dobivenog kontrakcijom vrhova. Korišten COL skup podataka.

Landmark A^*

- Najudaljeniji pohlepni pristup – problem računanja vrha najudaljenijeg od trenutnog skupa orijentacije. Rješenje: modificirana Dijkstrina pretraga s više početnih vrhova.
- Za vrh orijentacije l – za svaki $u \in V$ računanje vrijednosti: $d_G(l, u)$ (pretraga unaprijed) i $d_G(u, l)$ (pretraga unazad).
- Računanje vrijednosti funkcije π_t samo ukoliko je zaista potrebno.

	$t_{predproc}$	t_{pUpit}	$N_{pObradeno}$	Procjena
ALT-1	0.85 s	72.801 ms	108 954.04	54.63 %
ALT-2	1.44 s	68.549 ms	100 044.24	68.73 %
ALT-4	2.78 s	49.895 ms	64 907.34	74.73 %
ALT-8	5.38 s	31.819 ms	32 381.95	86.64 %
ALT-16	10.12 s	34.121 ms	23 657.20	88.13 %
ALT-24	18.95 s	37.683 ms	18 309.01	90.73 %
ALT-32	24.02 s	42.102 ms	16 888.85	91.41 %

Tablica: Odabir vrhova orijentacije: *slučajan način*. Skup podataka nad kojim je provedena usporedba je COL.

	$t_{predproc}$	t_{pUpit}	$N_{pObradeno}$	Procjena
ALT-1	1.88 s	84.902 ms	107 504.52	64.38 %
ALT-2	3.43 s	49.380 ms	66 754.56	74.93 %
ALT-4	6.74 s	31.579 ms	37 339.80	86.50 %
ALT-8	10.84 s	27.876 ms	27 601.40	90.44 %
ALT-16	19.48 s	34.789 ms	22 308.64	92.13 %
ALT-24	24.02 s	37.632 ms	18 563.00	93.25 %
ALT-32	35.81 s	43.861 ms	17 637.13	93.80 %

Tablica: Odabir vrhova orijentacije: *pohlepni najudaljeniji pristup*. Skup podataka nad kojim je provedena usporedba je COL.

Usporedba rezultata

- Usporedba i provjera duljine najkraćeg puta. Rekonstrukcija puta nije potrebna.
- Prioriteti vrhova CH algoritma – 2EDN.
- Broj vrhova orijentacije ALT algoritma iznosi 8.

	$t_{predproc}$	t_{pUpit}	$N_{pObrađeno}$
Dijkstra	0 s	94.460 ms	132 288.81
CH	13 min 58.97 s	4.348 ms	664.06
ALT	5.11 s	13.575 ms	12 989.65

Tablica: Usporedba algoritama na skupu podataka NY koji sadrži 264 346 vrhova i 730 100 bridova.

	$t_{predproc}$	t_{pUpit}	$N_{pObrađeno}$
Dijkstra	0 s	140.809 ms	222 503.23
CH	15 min 56.85 s	3.429 ms	444.11
ALT	10.84 s	27.876 ms	27 601.40

Tablica: Usporedba algoritama na skupu podataka COL koji sadrži 435 666 vrhova i 1 042 400 bridova.

	$t_{predproc}$	t_{pUpit}	$N_{pObrađeno}$
Dijkstra	0 s	375.670 ms	530 997.30
CH	2 h 39 min 17.98 s	8.842 ms	251.83
ALT	20.53 s	66.844 ms	62 278.29

Tablica: Usporedba algoritama na skupu podataka FLA koji sadrži 1 070 376 vrhova i 2 687 902 bridova.

	$t_{predproc}$	t_{pUpit}	$N_{pObrađeno}$
Dijkstra	0 s	741.203 ms	918 261.76
CH	4 h 50 min 28.70 s	10.59 ms	686.33
ALT	42.11 s	124.364 ms	103 718.06

Tablica: Usporedba algoritama na skupu podataka CAL koji sadrži 1 890 815 vrhova i 4 630 444 bridova.

Hvala na pažnji!