# Hopkroft-Karpov algoritam: projektni izvještaj

Mario Borna Mjertan, seminar iz kolegija *Oblikovanje i analiza algoritama*
Matematički odsjek, Prirodoslovno-matematički fakultet, Sveučilište u Zagrebu
ak. godina 2023./2024., nositelj: Matej Mihelčić, doc. dr. sc.

---

## Osnovno o problemu sparivanja

**Definicija** Sparivanje

**Sparivanje** u neusmjerenom grafu $G = (V, E)$ je podskup bridova $M \subseteq E$ takav da svaki vrh u $V$ ima najviše jedan s njim incidentan brid u $M$.

**Definicija** Maksimalno sparivanje

**Maksimalno sparivanje** u grafu je sparivanje maksimalne kardinalnosti. Drugim riječima, to je sparivanje u koje ne možemo dodati više nijedan brid, odnosno bilo koji nadskup njega više nije sparivanje.

**Definicija** Bipartitan graf

**Bipartitan graf** je graf $G = (V, E)$ čiji se vrhovi mogu particionirati u dva disjunktna skupa $V_1$ i $V_2$ tako da svaki brid iz $E$ spaja vrh iz $V_1$ s vrhom iz $V_2$.

Hoproft-Karpov algoritam daje način pronalaska maksimalnog sparivanja u bipartitnom grafu u $O(\sqrt{V}E)$. Jedan algoritam za pronalaska maksimalnog sparivanja već smo vidjeli u vidu Edmons-Karpova algoritma za pronalaženje maksimalnog toka, što je nešto općenitiji problem, u seminaru kolege Polančeca. Velik broj problema može se modelirati kao način pronalaska maksimalnog sparivanja u bipartitnom grafu - Cormen et al. izdvajaju, recimo, dogovaranje termina za intervjue za posao prema raspoloživosti intervjuera i kandidata. Osnovna ideja algoritma je inkrementalno povećavati veličinu sparivanja.

**Napomena**. Cormen et al. ne uvjetuju da se u *putu* u grafu ne ponavljaju čvorovi, odnosno definicija puta nije ekvivalentna onoj sa npr. Diskretne matematike. Utoliko naglašavam da je *jednostavan put* put bez ponavljanja čvorova, odnosno, *jednostavan put* je ekvivalentan pojam *putu* s Diskretne matematike. Analogna interpretacija se primjenjuje na pojam *jednostavnih ciklusa*.

### ⓘ *Definicija* $M$-alternirajuć put

Neka je $M$ sparivanje u neusmjerenom grafu $G = (V, E)$. $M$**-alternirajuć put** je jednostavan put čiji bridovi alterniraju između pripadanja $M$ i $E \backslash M$.

### ⓘ *Definicija* $M$-augmentirajuć put

Neka je $M$ sparivanje u neusmjerenom grafu $G = (V, E)$. $M$**-augmentirajuć put** je $M$-alternirajuć put koji počinje i završava u $E \backslash M$.

**Napomena**. Ovo znači da $M$-augmentirajuć put sadrži čvor više iz $E \backslash M$ nego iz $M$, pa mora imati neparno mnogo bridova.

**Napomena o notaciji**. Za proizvoljne skupove $A$ i $B$, s $A \triangle B$ označavamo simetričnu razliku $(A \backslash B) \cup (B \backslash A)$, a s $\mathscr{k}(A)$ kardinalnost skupa $A$.

# Dokaz korektnosti

### Lema 1

Neka je $M$ sparivanje u proizvoljnom neusmjerenom grafu $G = (V, E)$ i neka je $P$ $M$-augmentirajuć put. Tad je skup bridova $M' = M \triangle P$ također sparivanje u $G$ i $\mathscr{k}(M') = \mathscr{k}(M) + 1$.

**Dokaz**. Pretpostavimo da $P$ sadrži $q$ bridova tako da ih je $\lceil q/2 \rceil$ sadržano u $E \backslash M$ i $\lfloor q/2 \rfloor$ sadržano u $M$. Označimo tih $q$ vrhova s $(v_1, v_2), (v_2, v_3), \ldots, (v_q, v_{q+1})$. $P$ je $M$-augmentirajuć $\implies v_1$ i $v_{q+1}$ su nespareni s obzirom na $M$, ostali vrhovi u $P$ su spareni. Bridovi $(v_1, v_2), (v_3, v_4), \ldots (v_q, v_{q+1})$ pripadaju $E \backslash M$, a bridovi $(v_2, v_3), (v_4, v_5), \ldots, (v_{q-1}, v_q)$ pripadaju $M$. Primjenom simetrične razlike dobivamo da bridovi $(v_1, v_2), \ldots, (v_q, v_{q+1})$ pripadaju $M'$, dok $(v_2, v_3), \ldots, (v_{q-1}, v_q)$ pripadaju $E \backslash M'$. Svaki vrh $v_1, \ldots, v_q, v_{q+1}$ sparen je s obzirom na $M'$ te $M'$ dobiva dodatan brid u odnosu na $M$. Ostali bridovi i vrhovi u $G$ nisu zahvaćeni promjenom iz $M$ u $M'$. Slijedi da je $M'$ sparivanje u $G$ i $\mathscr{k}(M') = \mathscr{k}(M) + 1$. $\square$

### ✓ Korolar 2

Neka je $M$ sparivanje u proizvoljnom neusmjerenom grafu $G = (V, E)$, te neka su za neki $k \in \mathbb{N}$ s $P_i, i = 1, \ldots, k$ dani $M$-augmentirajući putevi u parovima disjunktnih vrhova. Tad je $M' = M \triangle (\cup_{i=1}^{k} P_i)$ sparivanje u $G$ te $\mathscr{k}(M') = \mathscr{k}(M) + k$.

**Dokaz**. Korištenjem asocijativnosti simetrične razlike i matematičkom indukcijom iz prethodne leme.

### Lema 3

Neka su $M$ i $M^*$ proizvoljna sparivanja u grafu $G = (V, E)$ te neka je $G' = (V, E')$ graf za $E' = M \triangle M^*$. Tad je $G'$ disjunktna unija jednostavnih puteva, jednostavnih ciklusa i/ili izoliranih vrhova. Bridovi u svakom takvom jednostavnom putu ili ciklusu alterniraju između $M$ i $M^*$. Ako je $\Bbbk(M^*) > \Bbbk(M)$, tad $G'$ sadrži barem $\Bbbk(M^*) - \Bbbk(M)$ vrhovno-disjunktnih $M$-augmentirajućih puteva.

**Dokaz**. Svaki vrh u $G'$ stupnja je najviše 2, budući da najviše dva brida iz $E'$ mogu biti incidentna s vrhom: najviše jedan iz $M$ i najviše jedan iz $M^*$.
$\implies$ Svaka komponenta povezanosti je ili izolirani vrh ili jednostavan put s bridovima koji alterniraju između $M$ i $M^*$ ili jednostavan ciklus parne duljine s bridovima koji alterniraju između $M$ i $M^*$.
Definicija $E'$ i $\Bbbk(M^*) > \Bbbk(M) \implies E'$ mora sadržavati $\Bbbk(M^*) - \Bbbk(M)$ više vrhova iz $M^*$ nego iz $M$.
Svaki ciklus u $G'$ ima paran broj bridova povučenih alternirajuće iz $M$ i $M^* \implies$ svaki ciklus ima jednako bridova iz $M$ i $M^*$
$\implies$ jednostavni putevi u $G'$ su razlog $\Bbbk(M^*) - \Bbbk(M)$ više bridova iz $M$
Svaki put koji sadrži različit broj bridova iz $M$ i $M^*$ pripada jednom od dva slučaja:

1. počinje i završava s bridovima iz $M \implies$ sadrži jedan rub više iz $M$
2. počinje i završava s bridovima iz $M^* \implies$ sadrži jedan rub više iz $M^*$
   $E'$ sadrži $\Bbbk(M^*) - \Bbbk(M)$ više rubova iz $M^*$ (nego iz $M$) $\implies$ barem je $\Bbbk(M^*) - \Bbbk(M)$ puteva iz slučaja 2, pri čemu je svaki od njih $M$-augmentirajuć. Budući da svaki vrh ima najviše dva incidentna brida iz $E'$, ti putevi moraju biti vrhovno disjunktni. □

### Korolar 4 (uvjet zaustavljanja)

Sparivanje $M$ u grafu $G = (V, E)$ je maksimalno sparivanje $\iff G$ ne sadržava $M$-augmentirajuć put.

**Dokaz**. Obratom po kontrapoziciji u oba smjera.
[ $\implies$ ]
Obrat po kontrapoziciji glasi: $G$ sadržava $M$-augmentirajuć put $\implies M$ nije maksimalno sparivanje u $G$.
Tvrdnja slijedi direktno iz Leme 1, definicije $M$-augmentirajućeg puta i maksimalnog sparivanja.
[ $\impliedby$ ]
Obrat po kontrapoziciji glasi: $M$ nije maksimalno sparivanje $\implies G$ sadržava $M$-augmentirajuć put.
Neka je $M^*$ maksimalno sparivanje takvo da je $\Bbbk(M^*) > \Bbbk(M)$. Tad iz Leme 3 direktno slijedi tvrdnja. □

### Hopcroft-Karpov algoritam

1. $M = \emptyset$
2. **ponavljaj**
3. > neka je $\mathcal{P} = P_1, \ldots, P_k$ maksimalan skup vrhovno-disjuktnih najkraćih $M$-augmentirajućih puteva
4. > $M = M \triangle \cup_{i=1}^{k} P_i$
5. **dok vrijedi** $P \neq \emptyset$
6. **vrati** $M$**

Korolarom 4 smo dokazali korektnost Hopcroft-Karpovog algoritma.

# Složenost

Hopkroft-Karp se može implementirati u $O(\sqrt{V}E)$ vremenu. Cormen et. al pokazuju da se petlja ponavlja $O(\sqrt{V})$ puta te predlažu implementaciju kojom je određivanje maksimalnog skupa vrhovno-disjunktnih najkraćih $M$-augmentirajućih puteva složenosti $O(E)$. Elegantan dokaz se može naći u *Introduction to Algorithms*.

Njihova implementacija se odvija u tri faze: formiranjem usmjerene verzije $G_M$ neusmjerena bipartitna grafa $G$, zatim stvaranjem usmjerenog acikličkog grafa $H$ iz $G_M$ varijantom BFS-a te konačno pronalaskom maksimalnog skupa vrhovno disjunktnih najkraćih $M$-augmentirajućih puteva varijatnom DFS-a na transpoziciji $H^T$ od $H$, pri čemu transpozicijom usmjerena grafa smatraju graf s obrnutim smjerovima svaka pojedinačna brida. Sličan pristup koristi i moja implementacija, a razlika je u tome što formira niz udaljenosti čvora od trenutna sparivanja u BFS-u određujući koji putevi se mogu naći kao najkraći $M$-augmentirajući te DFS-om pronalazi vrhovno disjunktan najkraći put.

# Empirijska analiza

Implementacija algoritma testirana je kroz dva parametrizirana testa na nenasumičnim grafovima. Kompilacija i izvršavanje s `cmake . && cmake --build . && ./main [t1_vc_max] [t1_iteration_count] [t1_logic] [t2_iteration_count] [t2_logic]`.

Aparatura za testiranja podržava dvije logike generiranja bridova:

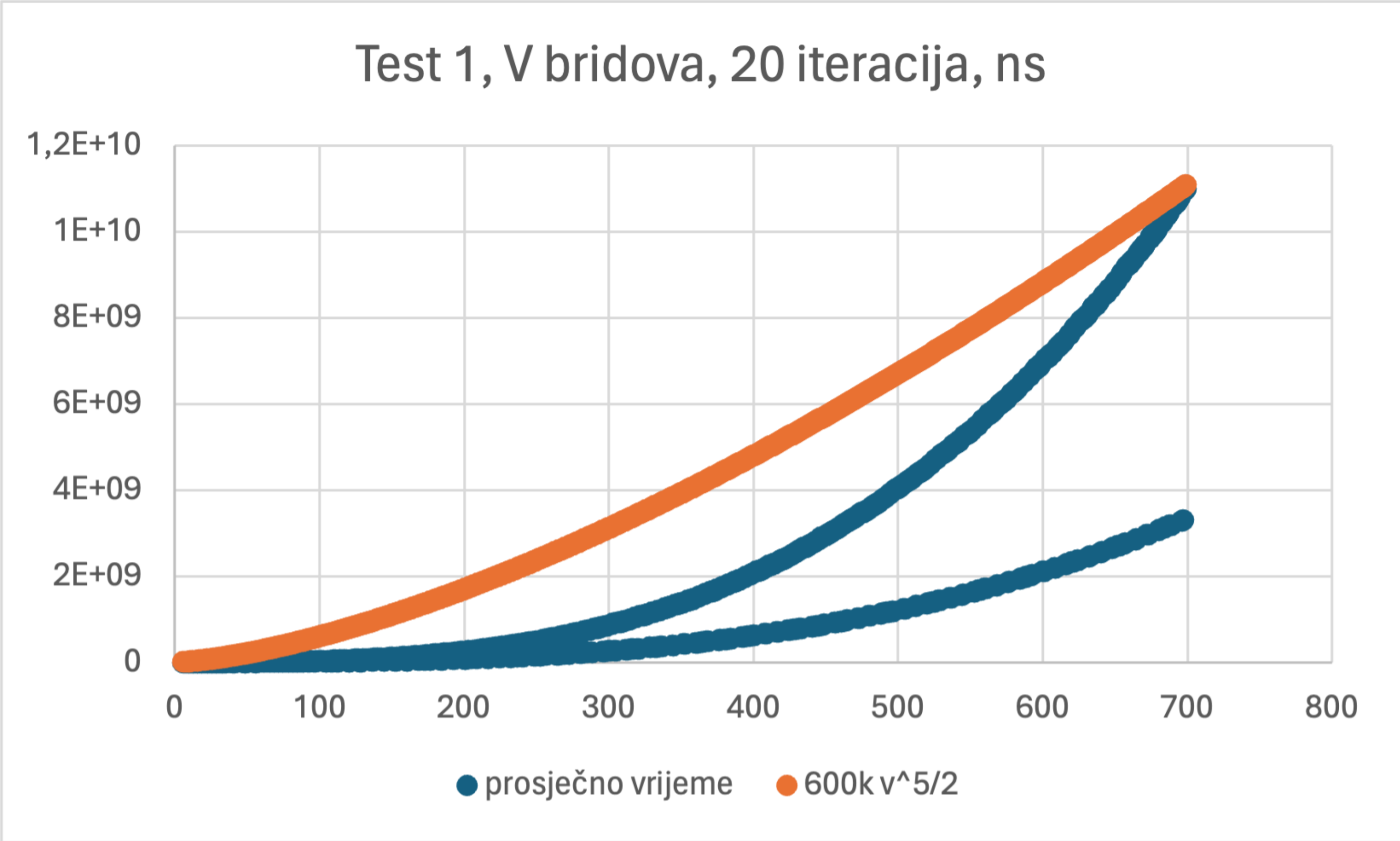
- logika 0, koja za $V = V_1 \cup V_2$ daje $E = V_1 \times V_2$ (simulira gust graf, što je loš slučaj za ovaj algoritam)
- logika 1, koja daje $k(E) = \lfloor\sqrt{V}\rfloor$ (relativno rijedak graf, što je dobar slučaj za ovaj algoritam)

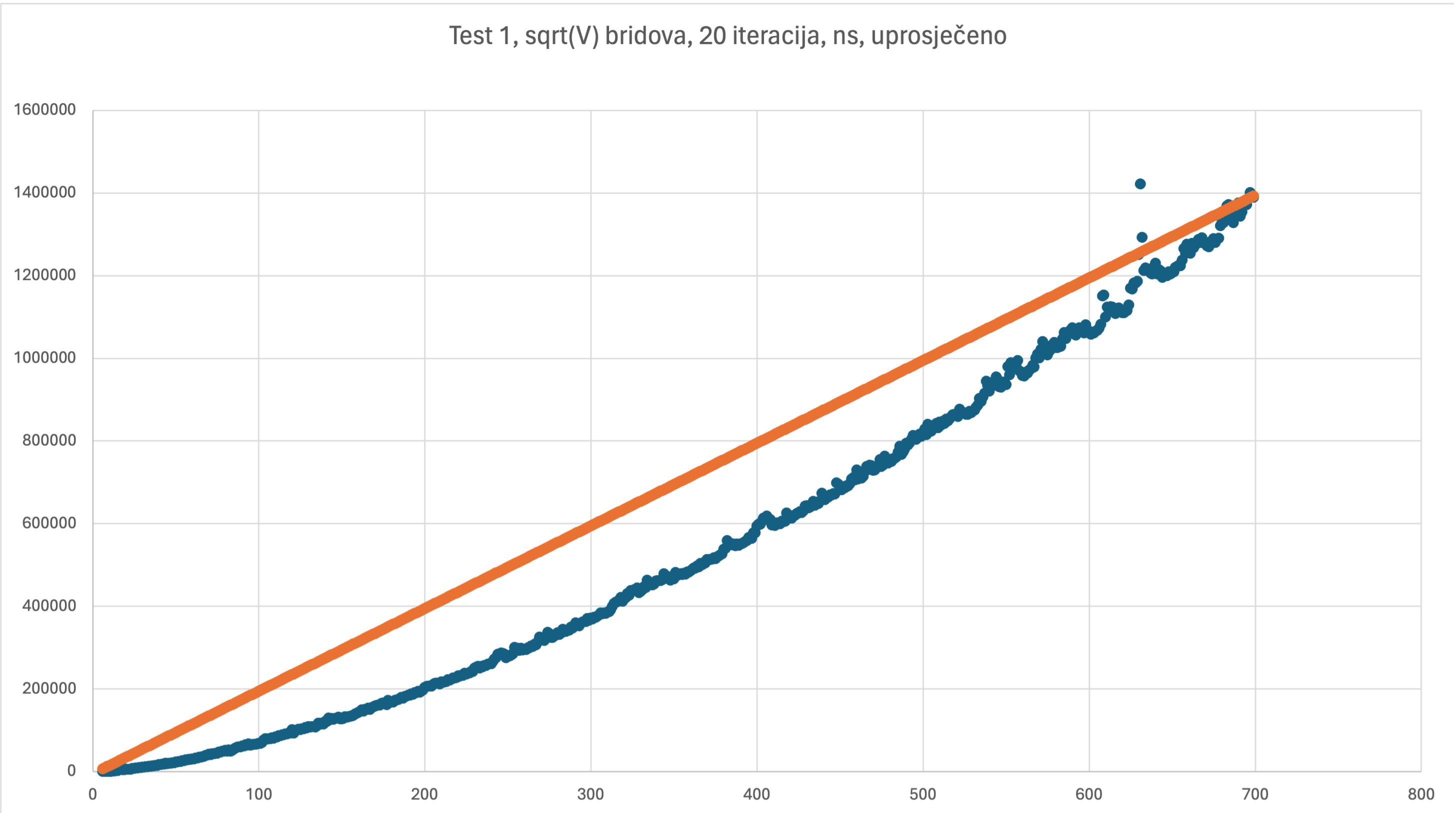
**Test 1** je parametriziran ograničenjem na broj vrhova `[t1_vc_max]` i logikom generiranja bridova `[t1_logic]`. Provodi testiranje na grafovima sa $6, 7, 8, \ldots, \mathrm{t1\_vc\_max}$ vrhova u danoj logici generiranja (što određuje broj bridova), ponavljajući svaki test `[t1_iteration_count]` puta. Ovaj test podržava `t1_logic: 2` vrijednost za testiranje obje logike.

**Test 2** je parametriziran brojem iteracija `[t2_iteration_count]` i logikom generiranja bridova `[t2_logic]`. Ne podržava vrijednost `t2_logic: 2`. Provodi testiranje na grafovima sa 650, 1000, 2500, 5000, 8000 i 10 000 vrhova u danoj logici generiranja.

Neobrađeni rezultati testova su priloženi u repozitoriju. Prema rezultatima iz prethodne točke, lagano dobivamo da u logici 0 (dopustimo si malo neformalnosti i označimo $\Bbbk(V) =: V$) očekujemo da vrijeme izvršavanja u najgorem slučaju iznosi $O(V^{\frac{5}{2}})$, odnosno $O(V)$ u logici 1. Na svim grafovima smatrat ćemo da $x$-os predstavlja broj vrhova grafa, a $y$-os prosječno vrijeme izvršavanja u naznačenim jedinicama. Također ćemo smatrati da je sve

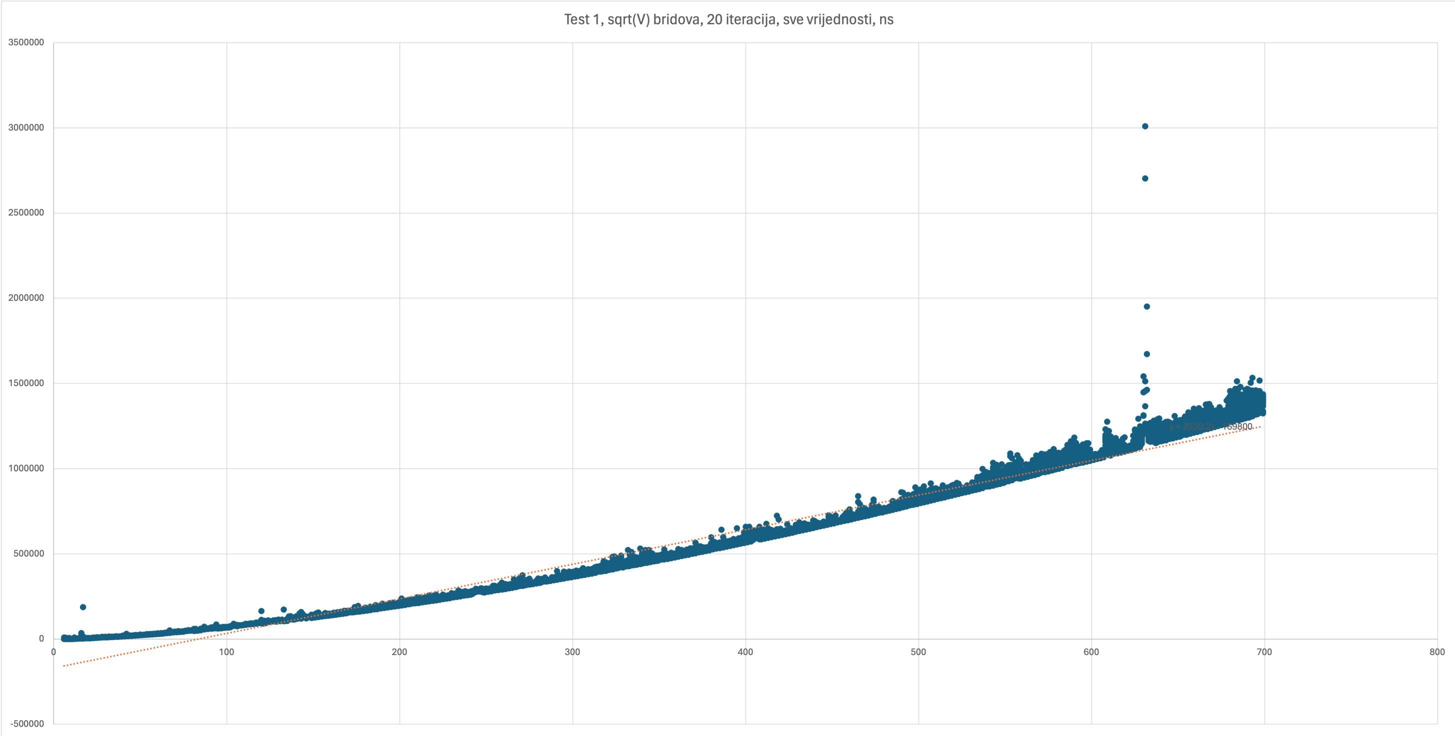
Procesor na kojem se provodilo testiranje je Apple M1 Pro. Na procesoru se dinamičko upravljanje frekvencijom ne može kontrolirati, a ista ovisno o jezgri i odlukama OS-a može varirati između 600 i 3220 MHz. U repozitoriju je priložen `powermetrics -s cpu_power -n 1` ispis. Pretpostavljam da je isto uzrok devijacijama (tj. prekidima u krivuljama prosječnog vremena odnosno činjenici da ih imam dvije ponegdje) koje vidimo na grafovima, budući da algoritamski uzrok tomu nisam našao i da ne vidim nikakva uzorka u prosječnu vremenu između točaka.



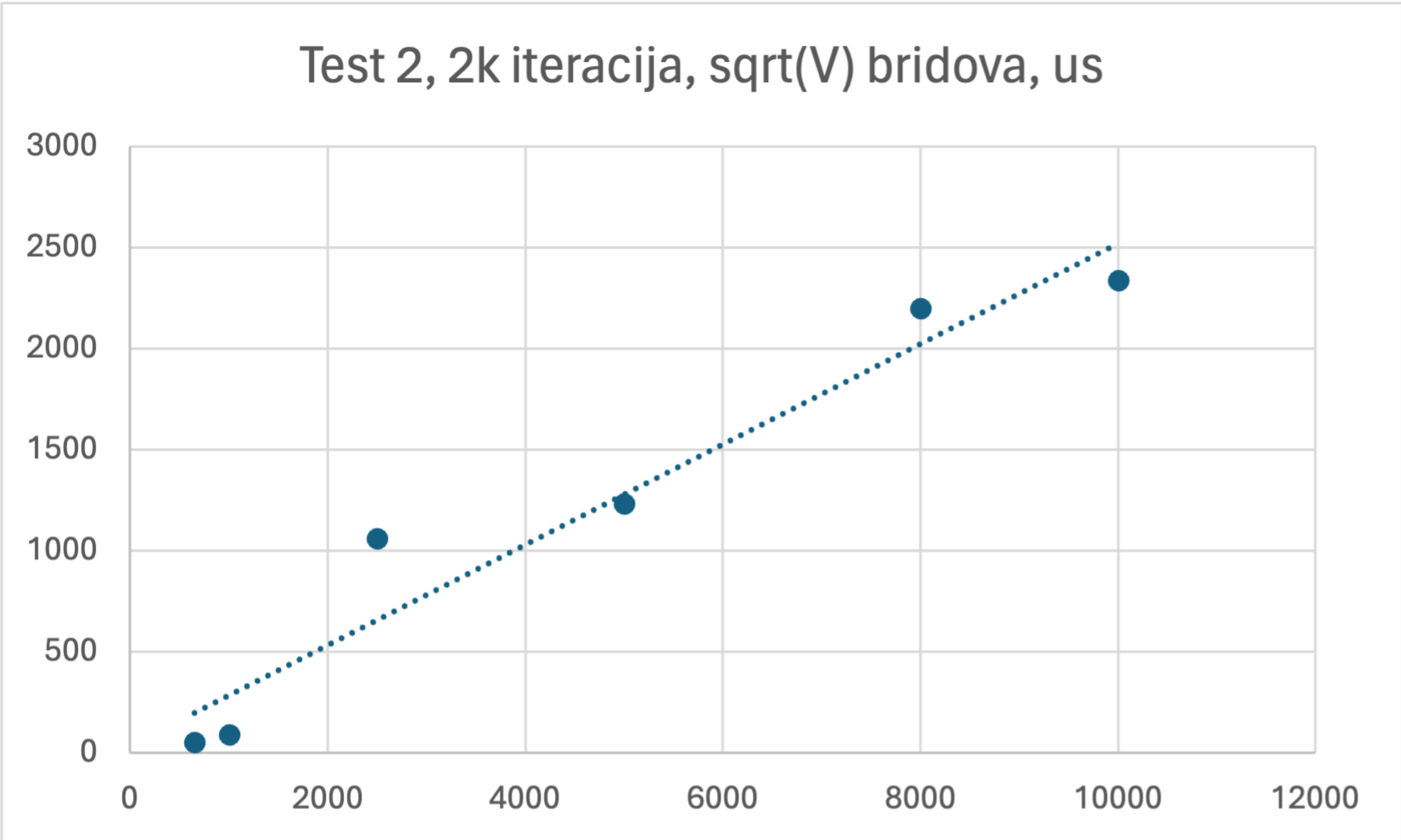




Oba grafa prikazuju prosječne vrijednosti za pojedini broj vrhova u logici 0, odnosno 1, respektivno u testu 1 za grafove s 6 do 700 vrhova na 20 iteracija. Iz njih možemo tvrditi da, do na outliere koji mogu biti uzrokovani vanjskim faktorima, u logici 0 doista imamo ogradu $O(V^{5/2})$ (ne nužno blisku), a u logici 1 imamo pravac.

Plotanjem svih vrijednosti u logici 1 na 20 iteracija nam se to i potvrđuje, doduše uz jasno vidljive outliere i varijaciju u brzini izvršavanja.

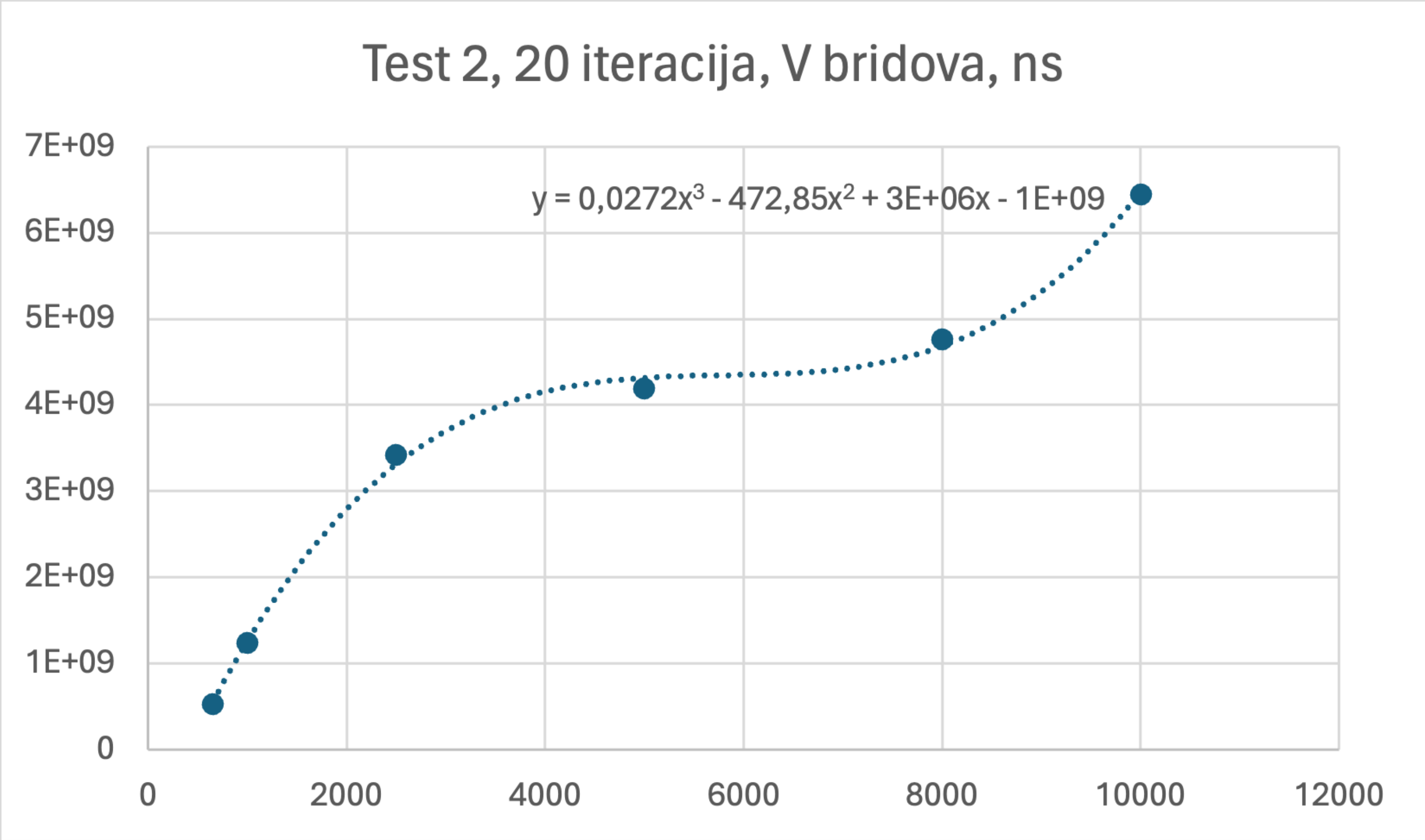



Analogno u testu 20 provedenom na 2000 iteracija u logici 2 dobivamo ovo za prosječne vrijednosti, što ponovno možemo opravdati s utjecajem operativnog sustava i procesorskog upravljanja energijom.

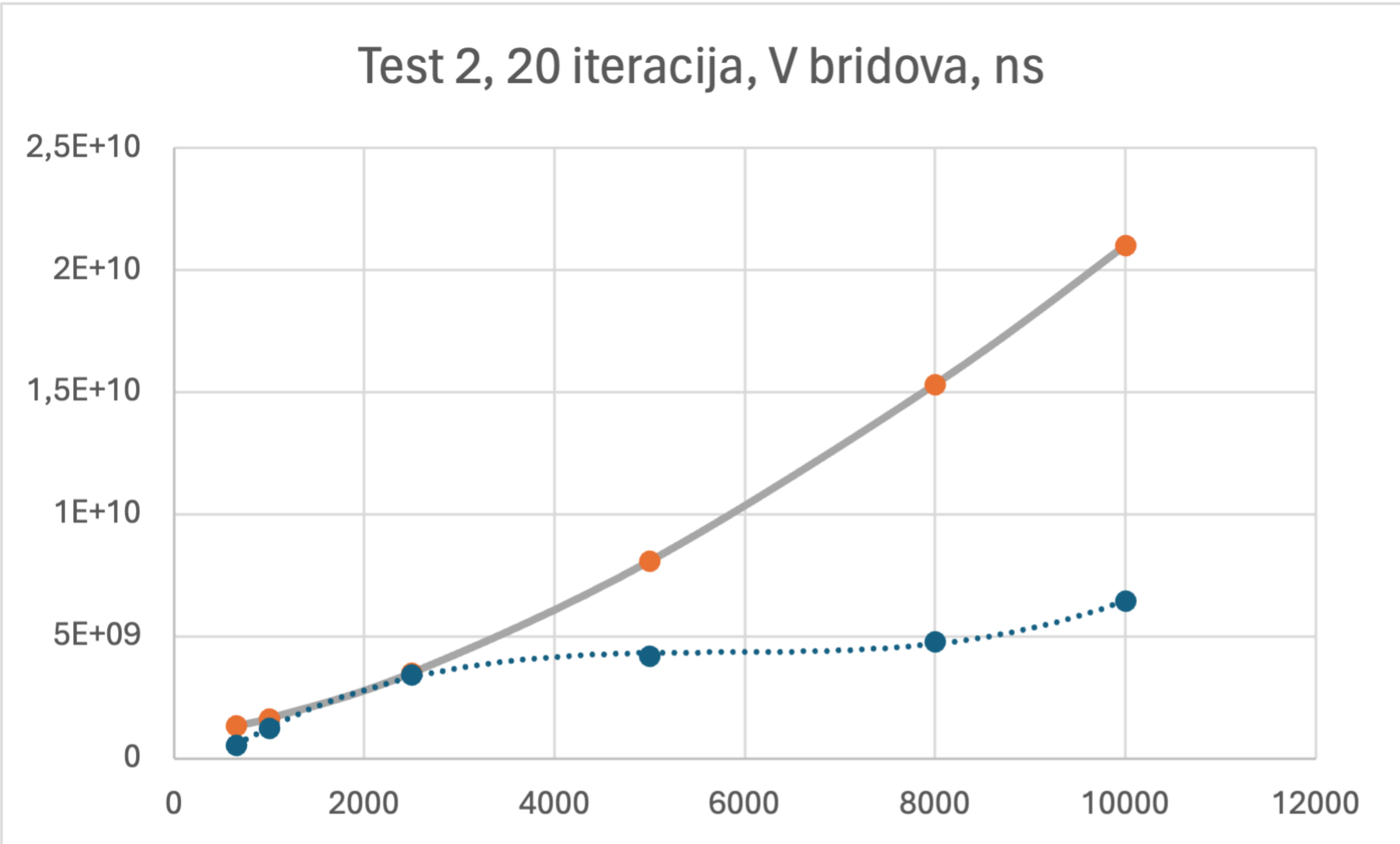



Provodeći test 2 u logici 1 na 20 iteracija vidimo da se isto poklapa s našim očekivanjem $O(V^{5/2})$
.

Kad interpoliramo točke u testu 2 polinomom stupnja 3, koeficijent uz $x^3$ je poprilično malen, pa nam to sugerira da ne trebamo odbaciti ocjenu koju smo dobili teorijski, odnosno da nemamo razloga vjerovati da $O(V^{5/2})$ nije dobra ograda.




Ako na isti graf dodamo npr. $20000V^2 + l$ imamo još jedan razlog za vjerovati da je ograda $O(V^{5/2}) u redu:




# Literatura


- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Cliffo: *Introduction to Algorithms (4th edition)*, 2022, The MIT Press

- Hopcroft-Karp algorithm, Wikipedia
- Bruno Polančec: *Edmonds-Karp algoritam*, seminar iz kolegija *Oblikovanje i analiza algoritama*, 2023./2024.
- Apple M1 Pro benchmarks, Notebookcheck.net i Find Clock Speed on M1 Mac, Apple Developer Forums u diskusiji oko linkova


Repozitorij ovog seminara: https://github.com/mbmjertan/oaa-hk