SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Mauro Raguzin

MEDIJAN-OD-TRI QUICKSORT

Projektni zadatak

Mentor:
doc. dr. sc. Goranka Nogo

Zagreb, siječanj 2022.

Sadržaj

Uvod

U ovom radu razmatramo inačicu algoritma Quicksort koja u svakoj glavnoj iteraciji odabire pivotni element kao medijan od tri slučajno odabrana elementa trenutnog (pod)niza realnih brojeva. Zatim dajemo dokaz faktora ubrzanja koje se ovim postiže povrh standardnog randomiziranog Quicksorta temeljenom na ukupnom broju usporedbi. Također, ukratko predstavljamo jednu generalizaciju ovog pristupa koja asimptotski omogućuje puno veća ubrzanja. Konačno, ukratko opisujemo popratni JavaScript program koji implementira ovaj algoritam kao i standardni Quicksort i omogućuje usporedbu vremena zidnog sata koje je potrebno da se njima sortira dugi nasumični niz realnih brojeva.

Poglavlje 1

Analiza vremenske složenosti

Već u prvim objavljenim radovima o Quicksort algoritmu nalazimo ideje njegovog poboljšanja kako bi se izbjegli patološki slučajevi u kojima se algoritam izvodi u kvadratnoj vremenskoj složenosti u odnosu na duljinu ulaznog polja. Tako Hoare [5] već u svom značajnom izvornom radu u kojem je Quicksort prvi put predstavljen navodi mogućnost korištenja malog uzorka trenutno promatranog segmenta polja u algoritmu radi određivanja novog pivotnog elementa kao medijana tog uzorka, no smatra da je teško procijeniti koje se točno smanjenje konstante vodećeg faktora vremenske složenosti može očekivati. Kasniji radovi poput [7] daju detaljnu analizu vremenske složenosti (i to ne samo u terminima broja usporedbi) posebno važnog slučaja gdje je taj uzorak veličine 3, ali i daljnjih generalizacija za veličinu uzorka $2k + 1$, $k \in \mathbb{N}$.

Mi se u ovom radu prvenstveno bavimo posebnim slučajem gdje je uzorak veličine 3, iako na kraju dajemo kratak uvid u jednu generalizaciju koja bi teoretski mogla dovesti do čak dvostruko većih ubrzanja standardnog Quicksorta.

Prvo dajemo najopćenitiju analizu ovog problema, iz koje ćemo lako moći zaključiti vremensku složenost za naš poseban slučaj u kojem se medijan odabire iz uzorka od samo 3 elementa; kasnije ćemo spomenuti i jedan puno jednostavniji argument koji vrijedi samo za ovaj naš slučaj.

1.1 Generalizirana analiza

Budući da nas zanima što točniji faktor ubrzanja ove inačice algoritma povrh standardnog randomiziranog Quicksorta, nije dovoljno raditi O-analizu u kojoj se zanemaruju konstante vodećeg faktora, već moramo detaljnije razmatrati rekurzivne relacije koje opisuju problem i riješiti ih u što više detalja tj. da nam rješenje otkrije bar vodeći faktor s pripadnom konstantom. Pritom moramo znati istu mjeru za standardni randomizirani Quicksort kako bismo mu mogli usporediti performanse s medijan-od-tri inačicom. Tijekom cijelog

rada pretpostavljamo da su svi brojevi u ulaznom polju međusobno različiti te da su sve permutacije tog polja jednako vjerojatne.

Iz literature je poznato [4] da je očekivani broj usporedbi standardnog Quicksorta (korištenjem Hoareove particijske sheme, no to nam neće biti važno u nastavku) na ulaznom polju duljine n jednak

$$2(n+1)\sum_{i=1}^{n}\frac{1}{i+1} - 2n \approx 1.39(n+1)\log_2 n - 2.85n + 2.15. \tag{1.1}$$

Kako bismo dobili sličnu vrijednost za medijan-od-tri inačicu, poopćavamo ovaj rezultat u oblik $\alpha n \log_2 n$, gdje α ne ovisi o n i izražava se na temelju vjerojatnosne distribucije pivotnog elementa. Pratimo izvod iz [3].

Korištenjem osnovne teorije informacija možemo formulirati problem pretraživanja kao problem sakupljanja informacija o pojedinoj permutaciji ulaznog polja, kao da želimo rekonstruirati originalno polje od konačno sortiranog. To znači da moramo dobiti onoliko novih informacija kolika je neodređenost sadržana u biranju nasumične permutacije polja. Poznato je da je u ovom diskretnom slučaju neodređenost jednaka entropiji diskretne distribucije $\{p_1, \ldots, p_{n!}\}$, gdje je $p_i = \frac{1}{n!}$ za $i = 1, \ldots, n!$ te je stoga jednaka

$$-\sum_{i=1}^{n!} p_i \log_2 p_i = \log_2(n!) \text{ bitova}, \tag{1.2}$$

što je općenito poznat rezultat. Također znamo da su odabiri permutacije lijevog i desnog podniza nakon particioniranja nezavisni te da je onda vjerojatnost odabira bilo koje permutacije u svakom odgovarajućem podnizu jednaka $\frac{1}{t!}$, gdje je t broj elemenata tog podniza.

Uz oznaku r za indeks odabranog pivotnog elementa, zaključujemo da je broj mogućih odabira permutacije nakon particioniranja jednak $r!(n-r)!$, a kako su odgovarajuće vjerojatnosti jednake $\frac{1}{r!}$ odnosno $\frac{1}{(n-r)!}$, imamo slično kao u (1.2) da je mjera neodređenosti $\log_2(r!(n-r)!)$. Ako sada uvedemo oznaku H za jediničnu entropiju tako da nH označava dobitak informacija ovim odabirom tj. smanjenje neodređenosti, dobivamo

$$nH = \log_2(n!) - \log_2(r!) - \log_2((n-r)!). \tag{1.3}$$

Sada želimo na neki način modelirati različite Quicksort varijante, a budući da se one sve bitno razlikuju samo po načinu odabira pivotnog elementa (pivota), dovoljno nam je uvesti novu slučajnu varijablu R pomoću koje možemo razmatrati različite distribucije pivota. Stavimo

$$P\{R = r\} = f_r,\ r = 1, \ldots, n,\ \sum_{r=1}^{n} f_r = 1. \tag{1.4}$$

U standardnom Quicksortu, pivot se uvijek odabire tako da $P\{R = k\} = \frac{1}{n},\ k = 1, \ldots, n$. Nas dakako zanima poopćenje za slučaj kada se pivot odabire kao medijan određenog uzorka segmenta, za što će biti potrebno odabrati odgovarajuću distribuciju.

Iz (1.4) sada možemo izračunati očekivani dobitak informacija particioniranjem kao

$$E(nH) = \log_2(n!) - \sum_{r=1}^{n} f_r(\log_2(r!) + \log_2((n-r)!)). \tag{1.5}$$

Asimptotski, za velike n i koristeći uobičajenu Stirlingovu aproksimaciju za $\log_2(n!)$, vrijedi

$$E(nH) \sim n\log_2 n - \sum_{r=1}^{n} f_r(r\log_2 r + (n-r)\log_2(n-r)) \tag{1.6}$$

i, zbog linearnosti očekivanja,

$$E(H) \sim -\log_2 n - \sum_{r=1}^{n} f_r\left(\frac{r}{n}\log_2 r + \frac{n-r}{n}\log_2(n-r)\right).$$

Sada uočimo da ovo možemo napisati na način koji će omogućiti logaritmu ispred sume da uđe u sumu i dovede cijeli izraz do tražene forme u kojoj je svaki sumand (bez f_r) oblika $m\log_2 m$ za neki m:

$$\begin{aligned} E(H) &\sim -\sum_{r=1}^{n}\left(\frac{r}{n} + \frac{n-r}{n}\right) f_r(-\log_2 n) - \sum_{r=1}^{n} f_r\left(\frac{r}{n}\log_2 r + \frac{n-r}{n}\log_2(n-r)\right) \\ &= -\sum_{r=1}^{n} f_r\left(\frac{r}{n}(\log_2 r - \log_2 n) + \frac{n-r}{n}(\log_2(n-r) - \log_2 n)\right). \end{aligned}$$

Konačno, dobivamo formulu

$$E(H) \sim -\sum_{r=1}^{n} f_r\left(\frac{r}{n}\log_2\left(\frac{r}{n}\right) + \frac{n-r}{n}\log_2\left(\frac{n-r}{n}\right)\right). \tag{1.7}$$

Za velike n se ova suma može aproksimirati integralom

$$E(H) \sim -\int_0^1 g(x)(x\log_2 x - (1-x)\log_2(1-x))\,\mathrm{d}x,$$

gdje je $g(x)$ vjerojatnosna funkcija gustoće varijable $X = \frac{R}{n}$. Ovaj integral je donekle moguće pojednostavniti ako se ograničimo samo na simetrične distribucije[1], što će se pokazati dostatnim za naš problem. Tada imamo integral

$$E(H) \sim -2\int_0^1 g(x)x\log_2 x\,\mathrm{d}x. \tag{1.8}$$

[1] To su one $g(x)$ za koje vrijedi $g(x) = g(1-x)$.

Uočimo da smo u dosadašnjoj analizi gledali na particioniranje niza od n elemenata kao osnovnu informacijsku jedinicu, no kao što smo uvodno napomenuli, uobičajeno je za to uzeti broj usporedbi elemenata s pivotom (kako bismo na kraju dobili vremensku složenost u ovisnosti o broju usporedbi), pa zbog gornje konstrukcije integrala i zbog činjenice da je uvijek potrebno napraviti ukupno točno n usporedbi za završiti particioniranje, imamo da se veličina $E(H)$ može tumačiti kao *prosječni* dobitak informacija dobiven *usporedbom*.

Koristeći (1.2) i uobičajenu Stirlingovu aproksimaciju dobivamo da je *ukupna* količina informacija koja se može dobiti sortiranjem polaznog polja duljine n jednaka $n \log_2 n$. Prema tome za *prosječan* broj potrebnih usporedbi (pa onda i za očekivano vrijeme izvođenja T_n) za sortiranje vrijedi

$$T_n \sim \frac{n \log_2 n}{E(H)} = \alpha n \log_2 n, \tag{1.9}$$

gdje je

$$\alpha = \left(- \int_0^1 g(x) x \log_2 x \, \mathrm{d}x \right)^{-1}.$$

[3]

Sada postaje moguće odrediti vremensku složenost medijan-od-tri Quicksorta korištenjem gornje formule i odgovarajuće funkcije $g(x)$, koju treba utvrditi za ovaj slučaj, te računanjem α. U tu svrhu koristimo jedan rezultat iz statistike koji kaže da, neovisno o distribuciji samih elemenata danog uzorka, njihove pozicije u sortiranom uzorku su jednoliko distribuirane. Također, poznato je da je vjerojatnosna funkcija gustoće medijana od uzorka veličine $2k + 1$ jednolike distribucije jednaka $g(x) = x^k(1 - x)^k / B(k + 1, k + 1)$, gdje je $B(x, y) = \int_0^1 t^{x-1}(t - 1)^{y-1} \, \mathrm{d}t$ beta funkcija[6]. Evidentno, $g(x)$ je simetrična distribucija.

Evaluiramo integral

$$\begin{aligned} &\int_0^1 g(x) x \ln x \, \mathrm{d}x \qquad (1.10) \\ &\quad = (1/B(k + 1, k + 1)) \int_0^1 x^{k+1} \ln x (1 - x)^k \, \mathrm{d}x \\ &\quad = (B(k + 2, k + 1)/B(k + 1, k + 1)) \cdot (\psi(k + 2) - \psi(2k + 3)) \\ &\quad = \frac{\psi(k + 2) - \psi(2k + 3)}{2}, \end{aligned}$$

gdje je

$$\psi(x) = \frac{\mathrm{d}}{\mathrm{d}x} \ln \Gamma(x).$$

Sada iz (1.9) i (1.10) izvodimo vrijednost tražene α za dani $k \in \mathbb{N}$ koji određuje veličinu uzorka $2k + 1$. Označimo je s α_k:

$$\begin{aligned}\frac{1}{\alpha_k} &= \frac{\psi(2k+3) - \psi(k+2)}{\ln 2} \\ &= \frac{\frac{1}{k+2} + \frac{1}{k+3} + \ldots + \frac{1}{2k+2}}{\ln 2} \\ &= \frac{1 - \frac{1}{2} + \frac{1}{3} - \ldots + \frac{1}{2k+1} - \frac{1}{2k+2}}{\ln 2}.\end{aligned}$$

U ovom raspisu smo koristili definiciju gama funkcije preko faktorijela te prije definirane njene logaritamske derivacije, jednostavan identitet

$$\frac{1}{n+1} + \frac{1}{n+2} + \ldots + \frac{1}{2n} = 1 - \frac{1}{2} + \ldots + \frac{1}{2n-1} - \frac{1}{2n},\ \forall n \in \mathbb{N}$$

lako provjerljiv matematičkom indukcijom te se sada još možemo pozvati na poznati rezultat iz analize da je $\sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n} = \ln 2$ kako bismo zaključili da $\lim_{k\to\infty} \alpha_k = 1$, što pokazuje da se ovim pristupom odabira pivota asimptotski približavamo teoretski minimalnoj vremenskoj složenosti internog sortiranja, iako cijena samog računanja medijana za tako velike uzorke u praksi postaje ograničavajući faktor zbog kojega se ovo ne isplati raditi bez neke vrste adaptivnosti [7].

Osim toga, možemo konkretno izračunati α_1 koji nas posebno zanima (α_0 samo potvrđuje (1.1)). Dobivamo $\alpha_1 = 1.188$, što je otprilike 15%-tno poboljšanje u odnosu na standardni randomizirani Quicksort.

1.2 Alternativna analiza

Do gotovo identičnog zaključka o vremenskoj složenosti medijan-od-tri Quicksorta kao u prošloj točki možemo doći puno kraćim, iako manje rigoroznim i manje općenitim argumentom koji je dio zadatka 7-5 u [2]. Autori tamo traže da se evaluira vjerojatnost odabira „dobrog" pivota za medijan-od-tri Quicksort, gdje je „dobar" pivot definiran kao onaj čija pozicija i u sortiranom polju zadovoljava $n/3 \leq i \leq 2n/3$. To možemo riješiti sumiranjem pojedinačnih vjerojatnosti za sve pozicije i koje zadovoljavaju taj uvjet, no prvo moramo izračunati vjerojatnost p_i da se pivot (medijan od tri) nalazi na i-toj poziciji sortiranog polja.

Odabirom bilo kojeg podskupa od 3 elementa polja imamo 6 mogućnosti njihovog uređaja i znamo da će se pod bilo kojim uređajem medijan nalaziti na srednjoj tj. drugoj poziciji tog uređenog podskupa. Dakle, moramo izračunati vjerojatnost da je srednji element odabranog (uređenog) podskupa na i-toj poziciji sortiranog polja, no kako za svaki

3-podskup postoji 6 permutacija, vjerojatnosti da je drugi element uređenog podskupa na i-toj poziciji sortiranog polja su jednake za sve moguće 3-podskupove. Konačno, za dobiti p_i dovoljno je odabrati jedan element lijevo od pivota i jedan element desno od pivota (koji je na poziciji i), pa imamo

$$p_i = \frac{(n-i)(i-1)}{\binom{n}{3}} = \frac{6(n-i)(i-1)}{n(n-1)(n-2)},$$

jer postoji $\binom{n}{3}$ načina za odabrati 3-podskup iz polja.

Sada se vraćamo na spomenuto sumiranje na zadanom intervalu u zadatku, koje odmah radi jednostavnosti zamjenjujemo integralom:

$$\sum_{i=n/3}^{2n/3} p_i \approx \int_{n/3}^{2n/3} \frac{6(n-x)(x-i)}{n(n-1)(n-2)} \mathrm{d}x,$$

što kad se raspiše i uzme limes od n u beskonačnosti daje $\frac{13}{27}$, što treba usporediti s istim integralom evaluiranim nad vjerojatnosti za standardni Quicksort, koja je jednostavno $p_i = \frac{1}{n}$, pa dobivamo slično kao iznad

$$\sum_{i=n/3}^{2n/3} \frac{1}{n} \approx \int_{n/3}^{2n/3} \frac{1}{n} \, \mathrm{d}n = \frac{1}{3},$$

što znači da, na temelju razlike dobivenih konstanti, opet imamo ubrzanje od otprilike 15%, pa se čini da spomenuti interval u [2] vrlo dobro aproksimira ponašanje odabira medijana na cijelom polju.

Poglavlje 2

Programsko rješenje

2.1 Opis programa

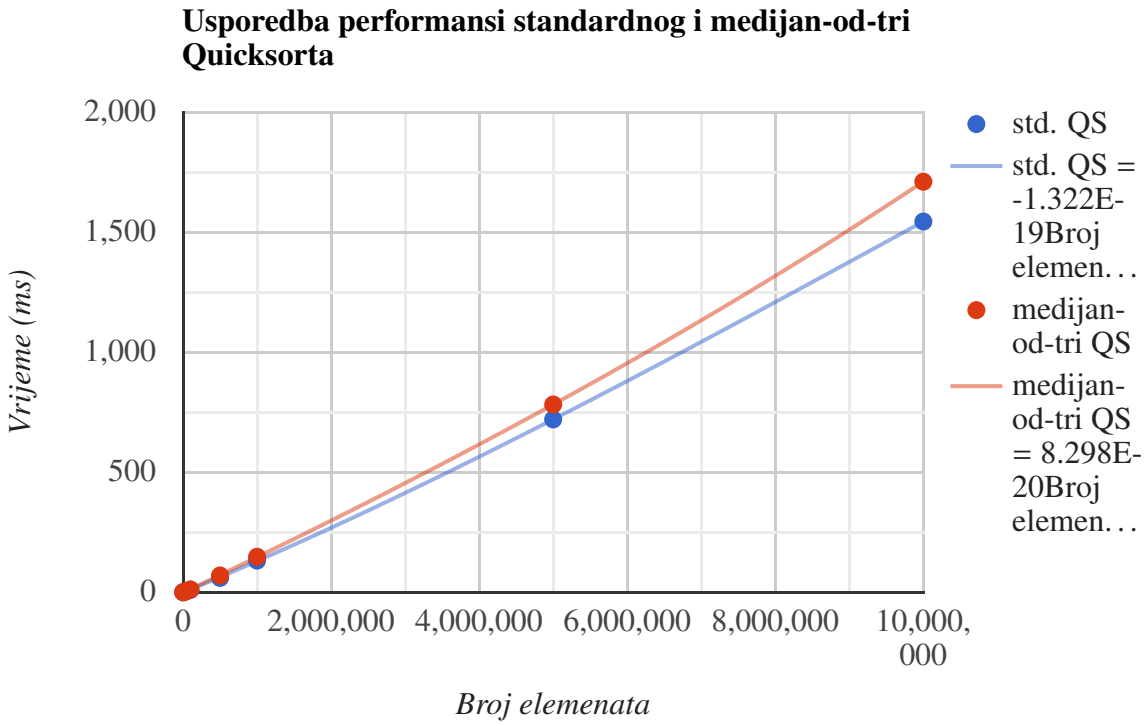
Popratni JavaScript program je jednostavna implementacija opisanog algoritma, bez ikakvih posebnih mikrooptimizacija. Osim medijan-od-tri Quicksorta, implementiran je i standardan randomizirani Quicksort i pritiskom na gumb `Sortiraj` korisnik dobiva, nakon izvjesnog vremena potrebnog i za generiranje dugog niza nasumičnih realnih brojeva i za sortiranje tim dvama algoritmima, tekst o proteklom vremenu zidnog sata za obavljanje prvo sortiranja standardnim Quicksortom, a zatim medijan-od-tri (to vrijeme ne uključuje vrijeme potrebno za generiranje polja; isto polje se dijeli za oba algoritma u danom pozivu). Neka razmatranja iz [1] su uzeta u obzir pri pisanju ovog programa.

Program se može jednostavno koristiti iz bilo kojeg Web preglednika. Kôd se nalazi u datoteci `qsort.js`.

2.2 Rezultati

Za kraj grafički predstavljamo dobivene rezultate u praksi. Za preciznije mjerenje vremena, napisan je još jedan program u C++-u, ekvivalentan iznad opisanom, kako bi se izbjegle moguće negativni utjecaji interpretera i sl. u okruženju Web preglednika.

Čak i s tom promjenom, dobivena vremena ne poklapaju se s obrađenom teorijom. Naime, ispada da ovako „ubrzan" algoritam ustvari *usporava* izvođenje sortiranja na jako velikim poljima na današnjim strojevima. To bi se moglo objasniti raznim faktorima, od kojih bi značajan mogao biti veličina i značaj *cache* memorije na današnjim procesorima, ključna za efikasno izvođenje programa i, u ovom slučaju najvažnije, brzo dohvaćanje vrijednosti iz polja. Budući da randomizirani medijan-od-tri radi odabira pivota zahtijeva dohvaćanje dva dodatna elementa povrh standardnog Quicksorta te još uz to odabire iste nasumično te



stoga može završiti „skačući" po cijelom polju, ovakav algoritam nije pogodan u pogledu očuvanja većine radnog dijela polja u relativno jako malom *cache*-u podataka.

Ispod se može vidjeti grafički prikaz trenda rasta vremena — nedvojbeno je da je medijan-od-tri u praksi sporiji. Još jedna varijanta medijan-od-tri Quicksorta, temeljena na odabiru medijana s uvijek istih indeksa, nije dala statistički značajno bolje rezultate od randomizirane medijan-od-tri varijante, što je u skladu s teorijom da je glavni problem veličina *cache*-a, jer činjenica što se fiksiraju pozicije ne znači mnogo kada su te pozicije, u vrlo velikim poljima, i dalje jako udaljene (iako sada ekvidistantne) jedna od druge.

Bibliografija

[1] Jon L Bentley i M Douglas McIlroy. „Engineering a sort function”. *Softw. Pract. Exp.* 23.11 (1993.), str. 1249–1265.

[2] Thomas H Cormen i dr. *Introduction to Algorithms*. 3. izdanje. London, England: MIT Press, 2009.

[3] M. H. van Emden. „Increasing the Efficiency of Quicksort”. *Commun. ACM* 13.9 (rujan 1970.), str. 563–567. ISSN: 0001-0782. DOI: 10.1145/362736.362753. URL: https://doi.org/10.1145/362736.362753.

[4] W. D. Frazer i A. C. McKellar. „Samplesort: A Sampling Approach to Minimal Storage Tree Sorting”. *J. ACM* 17.3 (srpanj 1970.), str. 496–507. ISSN: 0004-5411. DOI: 10.1145/321592.321600. URL: https://doi.org/10.1145/321592.321600.

[5] C. A. R. Hoare. „Quicksort”. *The Computer Journal* 5.1 (siječanj 1962.), str. 10–16. ISSN: 0010-4620. DOI: 10.1093/comjnl/5.1.10. eprint: https://academic.oup.com/comjnl/article-pdf/5/1/10/1111445/050010.pdf. URL: https://doi.org/10.1093/comjnl/5.1.10.

[6] Nikola Sarapa. *Teorija vjerojatnosti*. 3. izdanje. Zagreb, Hrvatska: Školska knjiga, 2002., str. 265–266.

[7] Robert Sedgewick. „The Analysis of Quicksort Programs”. *Acta Inf.* 7.4 (prosinac 1977.), str. 327–355. ISSN: 0001-5903. DOI: 10.1007/BF00289467. URL: https://doi.org/10.1007/BF00289467.

Sažetak

U ovom smo radu sagledali jednu od najstarijih varijanti originalnog Quicksort algoritma, medijan-od-tri Quicksort, koji bira pivotni element u svakoj glavnoj iteraciji sortiranja iz nasumično odabranog 3-podskupa ulaznog polja. Dali smo dokaz njegove prosječne vremenske složenosti (broja usporedbi), precizne do na konstantu vodećeg faktora te je usporedili sa složenosti standardnog (randomiziranog) Quicksorta i otkrili da je teorijsko ubrzanje oko 15%. Implementirali smo JavaScript i C++ program koji omogućuje interaktivnu usporedbu ova dva Quicksort algoritma na velikim, slučajno generiranim nizovima realnih brojeva. Konačno smo prikazali rezultate grafički te se uvjerili da je ovakav algoritam ipak u praksi sporiji na današnjim strojevima čak i od najobičnijeg Quicksorta, barem za velika polja i dali neke moguće razloge zašto je to tako.

Valja napomenuti da je teoretski moguće postići još veća ubrzanja Quicksorta ako se odabir veličine uzorka l učini *adaptivnim* tj. prilagodljivim n-u; rad [4] napominje da je najbolji oblik $l = 2^k - 1$ i da se njime, uz pretpostavku da je se za elemente uzorka biraju oni elementi polja koji su reprezentativni za cijelo polje tj. daju dobru procjenu kumulativne distribucije ulaznih vrijednosti. Tada je moguće pokazati da je ubrzanje povrh Quicksorta asimptotski jednako, teorijski (opet bez razmatranja utjecaja *cache*-a i ostalih faktora), skoro 30%.