SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Dokumentacija

Logaritamski dinamični *cuckoo* filter

Petra Habjanec i Andrea Milanović

Zagreb, svibanj, 2024.

Sadržaj

1. Uvod

U današnje vrijeme, s pojavom aplikacije koje obrađuju velike količine podataka, javlja se problem reprezentacije skupova. Organiziranjem elemenata skupa pomoću određene strukture podataka, reprezentacija skupa omogućuje pristup informacijama o elemenetima, podržavajući operacije kao što su umetanje, provjera članstva i brisnaje podataka. Vremenski i prostorno učinkovita struktura podataka za reprezentaciju skupa važna je u raznim primjenama aplikacija, kao što su pohrana na *cloud*-u, zaštita privatnosti, brojanje k-mera u sekvenciranju DNK i pretražuvanju mreže.

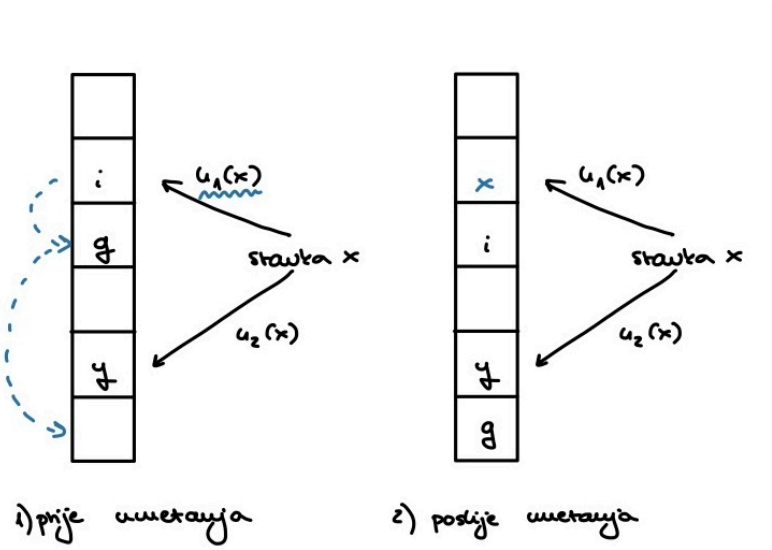
Logaritamski dinamični c*uckoo* filter predstavlja inovativno rješenje za ove izazove, nudeći poboljšane perfomanse kod vremena pristupa i utroška memorije. Ovaj algoritam koristi prednosti c*uckoo hash*-iranja s naprednin skupom podataka - binarnim stablom. Cilje je ove dokumentacije objasniti principe rada Logaritamskog dinamčnog c*uckoo* filtera, te našu implementaciju istog.

2. Logaritamski dinamički *cuckoo* filter

2.1. *Cuckoo hash* tablica

Cuckoo hash tablica[1] se sastoji od liste *l* kanti, svaka od kanti je jedinica u koju se pohranjuje stavka. Za svaku se stavku računaju dva neovisna *hash*-a koji označavaju dvije opcije kanti u koje možemo staviti stavku. Proces umetanja stavke izgleda ovako, gdje x označava stavku:

1) Tablica *cuckoo hash*-eva računa kandidate kanti adresa $h_1(x)$ i $h_2(x)$.
2) Ukoliko je bilo koja od $h_1(x)$ i $h_2(x)$ kanti prazna, spremamo stavku u praznu kantu, no ako ni jedna od njih nije prazna slučajno biramo jednu od njih, te umećemo stavku u nju.
3) Sada imamo jednu stavku koja je izbačena iz svog mjesta. Tu stavku zovemo žrtva te nju stavljamo u sebi alternativnu kantu.
4) Ukoliko je alternativna kanta prazna, stavka se stavlja u nju i proces je gotov, no ako nije vraćamo se na korak 3) te ga ponavljamo.[2]



Slika 1: Cuckoo hash tablica

Na slici 1 možemo vidjeti primjer umetanja stavke x u *cuckoo hash* tablicu. Vidimo da su obje moguće kante $h_1(x)$ i $h_2(x)$ pune te tako moramo slučajno jednu iz koje ćemo izbaciti stavku. U ovom slučaju biramo kantu $h_1(x)$ i tako izabacujemo stavku i, i tako postaje žrtva. Ona mora ići na svoje druge moguće mjesto u tablici, no na tom se mjesto nalazi stavka g. Na ovaj način g postaje sljedeća žrtva te se onda mora *preseliti* na svoju drugu opciju, koja je u ovom slučaju na sreću prazna. Tako dobivamo krajnje stanje tablice nakon umetanja stavke x.

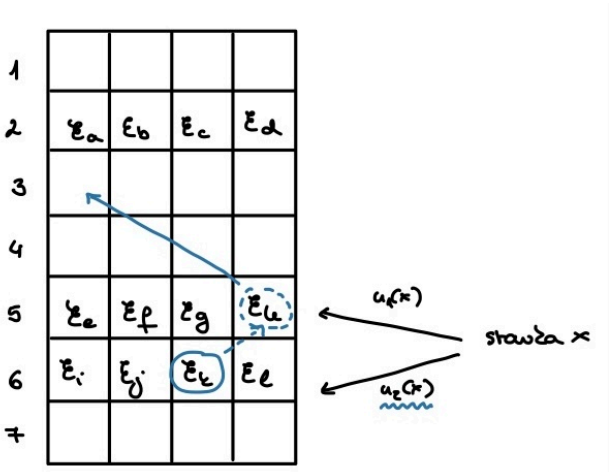
2.2. *Cuckoo* filter

Zamijenimo li originalnu stavku x s otiskom stavke(kasnije zvano ξ_x) dobivamo c*uckoo* filter. Tako preoblikovana *hash* tablica zauzima manje mjesta nego pohranjivanje samih stavaka. *Cuckoo* filter se sastoji od lista kanti veličine dužine *l*, svaka od kanti ima određen broj mjesta, *b*.

Kako ovdje je pohranjujemo same podatke već samo otiske podataka, javlja se problem pronalaženja alternativnih *hash*-eva za relociranje podataka. Kako bismo rješili ovaj problem, c*uckoo* filter se temelji na c*uckoo* *hash*-iranju djelomičnih ključeva. Takvim se *hash*-iranjem alternativna adresa računa XOR operacijom temeljenom na adresi trenutne kante i otiska.[3]

$$h_1(x) = hash(x)$$
$$h_2(x) = h_1(x) \oplus hash(_x)$$

Na slici 2 možemo vidjeti primjer umetanja u *cuckoo* filter. Na ovome gledamo u filter koji ima 7 kanti te svaka kanta ima zapremninu 4 stavke. Gledamo kako umećemo stavku x. Gledamo moguće kante $h_1(x)$ i $h_2(x)$, no obje kante su pune. Slučajnim odabirom biramo da ćemo stavku x umetnuti na $h_2(x)$, ali kako je ta kanta puna biramo jednu stavku koju ćemo izbaciti iz te kante, u ovom slučaju to je ξ_k, ta stavka sada postaje žrtva. ξ_k sada mora ići na svoje alternativno mjesto, ali je ta kanta već popunjena tako da i iz te kante moramo izbaciti slučajno odabranu stavku iz te kante. Sada bez mjesta ostaje ξ_h koji odlazi onda na svoje alternativno mjesto koje ima slobodnih mjesti u kanti. Tako da je sada proces umetanja u filter dovršen.

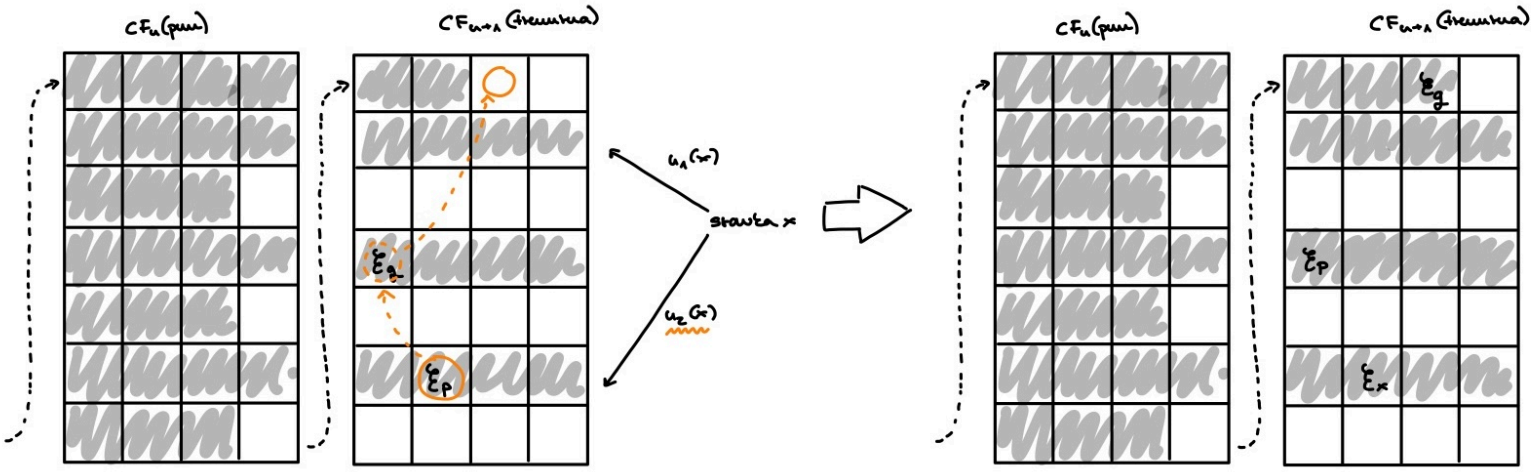


Slika 2: *Cuckoo* filter

2.3. Dinamični c*uckoo* filter

Kako bi se poduprijele velike količine podataka, smišljena je struktura dinamičnog c*uckoo* filtera, koji se sastoji od homogenih *cuckoo* filtera koji su povezani[4]. Svaki blok *cuckoo* filtera koristi brojač elemenata koji su pohranjeni u njemu, ukoliko je ta broj manji od predefiniranog kapaciteta taj se blok gleda kao aktivan. Kod umetanja u dinamični *cuckoo* filter prvo se pokušava umetnuti u prvi

aktivan blok, te ukoliko to ne uspije, dodaje se novi blok u kojeg se onda umeće nova stavka. Provjera pripadnosti dinamičnom *cuckoo* filteru se radi tako da prolazimo kroz listu te provjeravamo pripadnost otiska stavke svakog od blokova *cuckoo* filtera. Ukoliko prođemo sve blokove i ne pronađemo otisak koji odgovara otisku tražene stavke, funkcija vraća negativan odgovor. Ukoliko u bilom kojem od blokova u nizu pronađemo otisak koji odgovara otisku tražene stavke, funkcija vraća pozitivan odgovor. Brisanje se provodi prolaženjem kroz blokove te pokušajem brisanja u bloku, sve dok za jedan od blokova ne dobijemo pozivan odgovor da je izbrisan otisak koji odgovara otisku stavke koju pokušavamo obrisati.



Slika 3: Umetanje stavke u DCF

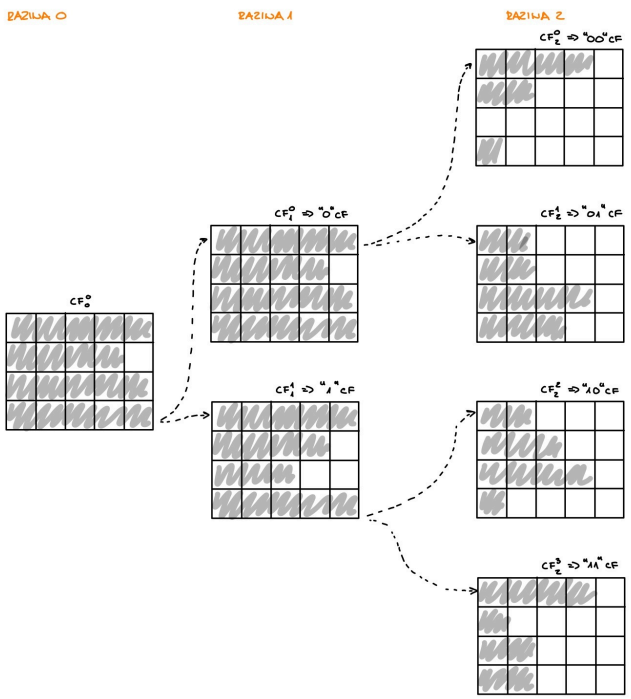
Na slici 3 možemo vidjeti primjer umetanja u dinamični *cuckoo* filter. Gdje umećemo stavku x, te ima izbor kanti $h_1(x)$ i $h_2(x)$, no obje kante na izbor su pune, tako da standardnim postupkom za *cuckoo* filter slučajno biramo između $h_1(x)$ i $h_2(x)$ - u ovome primjeru biramo $h_2(x)$. Tako da jednoga izbacujemo iz kante $h_2(x)$, u ovome je slučaju to ξ_p. ξ_p sada postaje žrtva te mora ići na svoje alternativno mjesto. Nažalost kanta s njegovim alternativnih mjestom je puna te se tako slučajnim odabirom ξ_q izbacuje iz kante, te on postaje žrtva i traži svoju alternativnu kantu. U njegovoj alternativnoj kanti ima mjesta te se smješta tamo i tako završava postupak.

2.4. Logaritamski dinamični *cuckoo* filter

Logaritamski dinamični *cuckoo* filter[2], u usporedbi s dinamičnim *cuckoo* filterom, koristi strukturu binarnog stabla. Takva struktura omogućava značajno nižu vremensku kompleksnost umetanja i provjere pripadnosti stavaka. Na slici 4 možemo vidjeti primjer izgleda LDCF-a.

Na početku LCDF započinje s jednim *cuckoo* filterom, u kojeg normalno umećemo otiske stavaka, sve dok jedno od umetanja ne uspije. Kada jedno umetanje ne uspije, zaključujemo da u taj filter više ne možemo umetati otiske stavaka. Ovisno o tome započinje li otisak stavke bitom nule ili jedinice, biramo hoće li taj otisak ići u novu lijevu ili desnu granu stabla. Nova je grana stabla isti cuckoo

filter kao i prijašnji no otisak koji se pohranjuje u njega je za jedan bit kreći. Kako određujemo poziciju u otiska u stablu ovisno o bitovima ostisaka, bitove kojima smo određivali poziciju u stablu ne moramo pohranjivati. Na taj način smanjujemo i memorijsko opterećenje našeg algoritma, pogotovo kod podataka velikih razmjera. Isto tako ovaj nam algoritam daje mogućnost korištenja dužih otisaka, kojima se smanjuju razmjeri lažnih negatnivih kod provjere pripadnosti.



Slika 4: Primjer izgleda LDCF-a

Kod provjere pripadnosti, na svakoj će razini samo jedan blok *cuckoo* filtera biti provjeren, te se tako jako smanjuje vremenska kompleksnost samog algoritma. U usporedbi s već prije spomenutim DCF algoritmom kojim ima vremensku kompleksnost $O(2^l - 1)$, LDCF $O(l)$. Time je vremenska složenost LDCF-a logaritamskom skalom. Iako LDCF ima veće vrijeme računanja, to je vrijeme za računanje mali trošak za platiti s tim da nemamo veliki broj selidba kod umetanja u blok *cuckoo* filtera[5].

2.4.1. Analiza točnosti, vremena izvođenja i utroška memorije

Svi su testovi izvršeni na LDCF-u s *cuckoo* filterima kapaciteta 20, kanti veličine 10 i maksimalnim brojem premještaja od 5. Duljine se otisaka stavaka mijenja kroz test primjere, što se može vidjeti i u tablicama.

E. coli podaci

Duljina genoma	Broj upisanih	Duljina otiska	k	Vrijeme umetanja [micro sekunde]	Vrijeme traženja [micro sekunde]	Vrijeme netočnih traženja(za 100 kmera) [micro sekunde]	Vrijeme brisanja [micro sekunde]	Memorija [kb]	Lažni pozitivni za umetanje
4641652	**464165**	16	10	1890776	1656359	1533	1655027	20780	100%
4641652	**464165**	18	10	1601956	1891586	613	1912019	20176	72%
4641652	**464165**	20	10	1563887	1970867	669	1959215	20240	33%
4641652	**464165**	22	10	1643220	1978216	657	1960902	20260	25%
4641652	**464165**	24	10	1735026	1972768	652	1957392	20248	21%
4641652	**464165**	26	10	1827584	1994579	671	1954796	20236	20%
4641652	**232082**	16	20	741297	849487	1485	849442	16864	99%
4641652	**232082**	18	20	735765	991181	693	978146	16704	54%
4641652	**232082**	20	20	729715	1011532	709	1001631	16840	24%
4641652	**232082**	22	20	770762	1010520	733	1003891	16852	7%
4641652	**232082**	24	20	806134	1013610	738	1006322	16868	5%
4641652	**232082**	26	20	856499	995871	694	1000149	16852	2%
4641652	**92833**	16	50	326414	460575	1707	461381	14844	78%
4641652	**92833**	18	50	311010	512144	952	509332	14792	28%
4641652	**92833**	20	50	316247	527642	946	514902	14688	10%
4641652	**92833**	22	50	317383	512337	974	512209	14660	3%
4641652	**92833**	24	50	327695	327695	942	510217	14692	3%
4641652	**92833**	26	50	341070	528682	941	512310	14668	4%
4641652	**46416**	16	100	189690	306729	2150	308428	14116	45%
4641652	**46416**	18	100	182198	331854	1541	332687	14104	18%
4641652	**46416**	20	100	181744	331512	1404	330382	14076	3%
4641652	**46416**	22	100	185066	330090	1372	328940	14080	2%
4641652	**46416**	24	100	191488	327872	1376	327843	14048	2%
4641652	**46416**	26	100	191712	326604	1364	326229	14076	4%
4641652	**23208**	16	200	109714	197939	3137	197815	13784	32%
4641652	**23208**	18	200	103731	212351	2094	214294	13760	7%
4641652	**23208**	20	200	111149	210917	2054	214462	13780	3%
4641652	**23208**	22	200	106129	208587	2042	208869	13732	1%
4641652	**23208**	24	200	109322	207421	2053	207298	13756	2%
4641652	**23208**	26	200	109433	208320	2019	208067	13696	3%

Tablica 1: Rezultati testa za genom E. Coli

Duljina genoma	Broj upisanih	Duljina otiska	k	Vrijeme umetanja [micro sekunde]	Vrijeme traženja [micro sekunde]	Vrijeme netočnih traženja(za 100 kmera) [micro sekunde]	Vrijeme brisanja [micro sekunde]	Memorija [kb]	Lažni pozitivni za umetanje
1000	100	16	10	44	36	296	36	2704	0%
1000	100	18	10	43	36	162	35	2704	0%
1000	100	20	10	43	36	164	35	2404	0%
1000	100	22	10	43	37	161	36	2704	0%
1000	100	24	10	44	35	166	34	2720	0%
1000	100	26	10	44	38	169	37	2724	0%
1000	50	16	20	21	19	399	19	2704	0%
1000	50	18	20	20	18	186	18	2704	0%
1000	50	20	20	21	18	187	18	2704	0%
1000	50	22	20	21	18	185	18	2692	0%
1000	50	24	20	21	18	188	18	2724	0%
1000	50	26	20	21	18	185	18	2704	0%
1000	20	16	50	16	12	604	13	2704	0%
1000	20	18	50	13	12	337	12	2704	0%
1000	20	20	50	13	12	347	12	2704	0%
1000	20	22	50	14	12	330	12	2724	0%
1000	20	24	50	14	12	330	12	2704	0%
1000	20	26	50	15	12	352	12	2704	0%
1000	10	16	100	10	8	808	8	2704	0%
1000	10	18	100	11	8	644	9	2704	0%
1000	10	20	100	11	9	606	9	2704	0%
1000	10	22	100	11	8	579	8	2704	0%
1000	10	24	100	10	8	578	8	2720	0%
1000	10	26	100	10	8	581	8	2704	0%
1000	5	16	200	7	6	1381	6	2704	0%
1000	5	18	200	8	6	1069	5	2704	0%
1000	5	20	200	7	6	1050	5	2704	0%
1000	5	22	200	7	6	1037	5	2724	0%
1000	5	24	200	8	6	1040	5	2704	0%
1000	5	26	200	8	6	1069	6	2744	0%

Tablica 2: Rezultati testa za simulirane podatke veličine 10^3

Duljina genoma	Broj upisanih	Duljina otiska	k	Vrijeme umetanja [micro sekunde]	Vrijeme traženja [micro sekunde]	Vrijeme netočnih traženja(za 100 kmera) [micro sekunde]	Vrijeme brisanja [micro sekunde]	Memorija [kb]	Lažni pozitivni za umetanje
10000	1000	16	10	921	961	421	951	2788	2%
10000	1000	18	10	970	1005	250	999	2748	1%
10000	1000	20	10	1132	1124	310	1020	2748	0%
10000	1000	22	10	1023	1023	264	1017	2768	0%
10000	1000	24	10	1039	995	241	1008	2748	1%
10000	1000	26	10	984	1071	321	999	2796	0%
10000	500	16	20	379	375	522	363	2760	0%
10000	500	18	20	382	368	255	376	2760	0%
10000	500	20	20	383	357	252	372	2740	0%
10000	500	22	20	383	360	249	360	2740	0%
10000	500	24	20	379	370	263	382	2760	0%
10000	500	26	20	388	361	245	361	2740	0%
10000	200	16	50	170	154	626	151	2736	1%
10000	200	18	50	166	141	381	140	2736	1%
10000	200	20	50	166	141	390	139	2736	0%
10000	200	22	50	166	141	390	140	2756	0%
10000	200	24	50	166	140	385	140	2736	1%
10000	200	26	50	166	143	390	141	2756	1%
10000	100	16	100	86	81	791	83	2736	1%
10000	100	18	100	92	87	606	85	2736	1%
10000	100	20	100	91	84	601	84	2736	1%
10000	100	22	100	87	82	577	82	2756	1%
10000	100	24	100	89	82	575	82	2736	1%
10000	100	26	100	86	81	593	81	2748	1%
10000	50	16	200	62	59	1316	60	2752	0%
10000	50	18	200	59	56	1056	56	2736	0%
10000	50	20	200	59	57	1036	56	2736	0%
10000	50	22	200	59	56	1036	56	2736	0%
10000	50	24	200	60	58	1078	114	2736	0%
10000	50	26	200	59	56	1037	59	2736	0%

Tablica 3: Rezultati testa za simulirane podatke veličine 10^4

Duljina genoma	Broj upisanih	Duljina otiska	k	Vrijeme umetanja [micro sekunde]	Vrijeme traženja [micro sekunde]	Vrijeme netočnih traženja(za 100 kmera) [micro sekunde]	Vrijeme brisanja [micro sekunde]	Memorija [kb]	Lažni pozitivni za umetanje
100000	10000	16	10	17255	20604	653	20330	3320	9%
100000	10000	18	10	17068	22809	455	22158	3260	2%
100000	10000	20	10	18427	22025	419	24373	3260	0%
100000	10000	22	10	20717	26250	447	22762	3252	2%
100000	10000	24	10	19638	21972	424	22824	3256	0%
100000	10000	26	10	20048	22163	421	22217	3260	1%
100000	5000	16	20	7358	9356	710	9359	3180	5%
100000	5000	18	20	7836	9851	504	10544	3184	1%
100000	5000	20	20	7515	10062	433	9748	3180	1%
100000	5000	22	20	8830	9821	425	10109	3188	0%
100000	5000	24	20	8419	10085	455	9803	3188	1%
100000	5000	26	20	8455	9725	426	9863	3184	1%
100000	2000	16	50	3333	3993	861	3967	3136	1%
100000	2000	18	50	3219	4456	626	4666	3144	1%
100000	2000	20	50	3164	4680	565	4318	3140	0%
100000	2000	22	50	3561	4362	550	4692	3176	0%
100000	2000	24	50	3556	4593	602	4448	3140	0%
100000	2000	26	50	3388	4322	540	4693	3140	0%
100000	1000	16	100	1565	2140	1141	2274	3128	0%
100000	1000	18	100	1563	2407	855	2290	3124	0%
100000	1000	20	100	1649	2346	850	2410	3128	0%
100000	1000	22	100	1601	2279	825	2271	3124	0%
100000	1000	24	100	1610	2285	822	2276	3124	0%
100000	1000	26	100	1574	2296	820	2302	3120	0%
100000	500	16	200	846	1135	1503	1145	3156	0%
100000	500	18	200	847	1138	1285	1152	3116	0%
100000	500	20	200	875	1140	1272	1143	3116	0%
100000	500	22	200	869	1126	1248	1134	3116	0%
100000	500	24	200	888	1149	1332	1162	3168	0%
100000	500	26	200	864	1127	1262	1123	3116	0%

Tablica 4: Rezultati testa za simulirane podatke veličine 10^5

Duljina genoma	Broj upisanih	Duljina otiska	k	Vrijeme umetanja [micro sekunde]	Vrijeme traženja [micro sekunde]	Vrijeme netočnih traženja(za 100 kmera) [micro sekunde]	Vrijeme brisanja [micro sekunde]	Memorija [kb]	Lažni pozitivni za umetanje
1000000	100000	16	10	279984	313895	1298	312818	7804	81%
1000000	100000	18	10	253853	345112	571	344280	7684	33%
1000000	100000	20	10	254432	355181	586	360430	7636	12%
1000000	100000	22	10	265451	347056	566	345136	7596	3%
1000000	100000	24	10	280541	344877	555	343038	7596	1%
1000000	100000	26	10	295310	345895	634	343369	7628	2%
1000000	50000	16	20	129979	157975	856	155659	7008	63%
1000000	50000	18	20	118649	167927	631	168643	6976	16%
1000000	50000	20	20	124034	180034	625	170292	6924	6%
1000000	50000	22	20	126532	166380	612	165776	6948	1%
1000000	50000	24	20	130279	167634	589	165541	6948	0%
1000000	50000	26	20	139260	164487	580	164756	6908	2%
1000000	20000	16	50	51969	78737	1639	78589	6508	30%
1000000	20000	18	50	52562	85393	840	85286	6528	11%
1000000	20000	20	50	55529	86985	827	92212	6552	4%
1000000	20000	22	50	56435	83496	822	82745	6516	3%
1000000	20000	24	50	59525	86249	826	83272	6560	1%
1000000	20000	26	50	61642	97110	861	101064	6524	6%
1000000	10000	16	100	29378	49485	2085	50090	6338	11%
1000000	10000	18	100	29611	52106	1187	53366	6400	4%
1000000	10000	20	100	29397	52262	1213	52335	6368	2%
1000000	10000	22	100	30549	51125	1161	52243	6404	0%
1000000	10000	24	100	31866	51345	1157	52014	6360	1%
1000000	10000	26	100	32566	50974	1172	55608	6408	1%
1000000	5000	16	200	16394	28826	2309	28978	6284	7%
1000000	5000	18	200	16491	31448	1820	32147	6276	3%
1000000	5000	20	200	16370	31277	1747	31285	6284	2%
1000000	5000	22	200	18109	32017	1710	31059	6320	2%
1000000	5000	24	200	16718	31183	1735	31252	6328	0%
1000000	5000	26	200	23752	31668	1769	31561	6292	0%

Tablica 5: Rezultati testa za simulirane podatke veličine 10^6

Duljina genoma	Broj upisanih	Duljina otiska	k	Vrijeme umetanja [micro sekunde]	Vrijeme traženja [micro sekunde]	Vrijeme netočnih traženja(za 100 kmera) [micro sekunde]	Vrijeme brisanja [micro sekunde]	Memorija [kb]	Lažni pozitivni za umetanje
10000000	1000000	16	10	4091922	3630272	1322	3608837	48316	100%
10000000	1000000	18	10	3966443	4282278	1411	4281142	48808	83%
10000000	1000000	20	10	3916028	4646981	940	4653876	49256	34%
10000000	1000000	22	10	4081888	4741478	1511	4730522	49392	12%
10000000	1000000	24	10	4305650	4769876	1552	4739641	49400	3%
10000000	1000000	26	10	4532844	4740551	1627	4738424	49380	4%
10000000	500000	16	20	1749072	2121659	1373	1909077	41824	99%
10000000	500000	18	20	1790129	2276557	1001	2273136	42020	77%
10000000	500000	20	20	1815928	2423107	767	2401060	42256	29%
10000000	500000	22	20	1894082	2449740	794	2438458	42252	11%
10000000	500000	24	20	2000048	2444259	802	2428373	42256	4%
10000000	500000	26	20	2121659	2444259	775	2432115	42276	3%
10000000	200000	16	50	801828	1062667	1122	1077318	37704	95%
10000000	200000	18	50	735411	1215810	1010	1212947	37484	53%
10000000	200000	20	50	737619	1251875	999	1255870	37324	19%
10000000	200000	22	50	767734	1247947	1032	1246276	37308	8%
10000000	200000	24	50	793320	1244444	1022	1253073	37320	4%
10000000	200000	26	50	825086	1243122	1027	1240896	37272	5%
10000000	100000	16	100	460772	733518	2228	737666	36148	77%
10000000	100000	18	100	439522	809693	1578	806844	36028	22%
10000000	100000	20	100	433173	821222	1507	845699	35956	9%
10000000	100000	22	100	447246	837099	1509	830409	35952	2%
10000000	100000	24	100	459600	816993	1476	809781	35972	1%
10000000	100000	26	100	479732	806435	1464	804510	35944	1%
10000000	50000	16	200	273495	491812	3178	497774	35416	52%
10000000	50000	18	200	257845	527078	2300	528920	35296	16%
10000000	50000	20	200	257579	526735	2216	527753	36206	3%
10000000	50000	22	200	260354	514159	2163	513671	35296	5%
10000000	50000	24	200	262396	508914	2304	510424	35256	4%
10000000	50000	26	200	273419	511378	2158	509859	35288	1%

Tablica 6: Rezultati testa za simulirane podatke veličine 10^7

3. Zaključak

U ovome smo radu sagledali dinamični *cuckoo* filter kao inovativno rješenje za efikasnu reprezentaciju velikih skupova podataka. U uvodnom dijelu, postavljamo temelje razumijevanja problematike velikih podataka i potrebe za vremenski i prostorno učinkovitim algoritmima i strukturama podataka.

Prolaskom kroz razvoj od osnovne cuckoo hash tablice, preko cuckoo filtera i dinamičnog cuckoo filtera, došli smo do LDCF-a, koji predstavlja značajan napredak u smislu performansi. Korištenjem binarnog stabla u strukturi LDCF-a, postignuta je smanjena vremenska složenost operacija umetanja i provjere pripadnosti.

Kod analize točnosti, vremenske i prostorne učinkovitosti možemo jasno vidjeti da se očitavaju prednosti LDCF-a, s povećanjem duljine otiska, smanjuje se postotak lažnih pozitivnih, no zauzeta memorija se previše ne povećava.

4. Literatura

[1] R. Pagh and F. F. Rodler, “Cuckoo hashing,” Journal of Algo- rithms, vol. 51, no. 2, pp. 122–144, 2004.

[2] F. Zhang, H. Chen, H. Jin, and P. Reviriego, "The Logarithmic Dynamic Cuckoo Filter," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, Wuhan, China, 2021, pp. 948-959. Huazhong University of Science and Technology, National Engineering Research Center for Big Data Technology and System, Cluster and Grid Computing Lab, Services Computing Technology and System Lab, School of Computer Science and Technology; Universidad Carlos III de Madrid, Departamento de Ingeniería Telemática.

[3] B. Fan, D. G. Andersen, M. Kaminsky, and M. Mitzenmacher, “Cuckoo filter: Practically better than bloom,” in Proceedings of CoNEXT, 2014.

[4] H. Chen, L. Liao, H. Jin, and J. Wu, “The dynamic cuckoo filter,” in Proceedings of ICNP, 2017.

[5] F. Wang, H. Chen, L. Liao, F. Zhang, and H. Jin, “The power of better choice: Reducing relocations in cuckoo filter,” in Proceedings of ICDCS, 2019.