

FACULTAT D'INFORMÀTICA DE BARCELONA

ALGORÍSMIA

Pràctica de Hashing

Josep Ciurana Herrera (josep.ciurana) Oriol Closa Márquez (oriol.closa)

Sumari

1	Intro	ducció																				2
2	Cerc	a binàr	ia										 									3
3	Hasi	<i>h</i> amb ta	aula										 				 					4
	3.1	Open h	nashing										 									4
		3.1.1	Linear p	orobii	ng								 									4
		3.1.2	Quadra	tic pr	obin	a .							 									4
		3.1.3	Double	•		_																4
		3.1.4	Cuckoo		-																	5
	3.2	Separa	ite chaini		J																	5
	·-	321	Move to	_																		5
		3.2.2	Exact fit																			5
4	Filtr	e de Blo	oom										 									6
5	Exp	eriment	s										 									7
	5.1	Cerca	binària										 									8
	5.2	Hash a	ımb taula										 									9
	5.3		e Bloom																			12
Bi	bliogr	afia											 									13
A	Dades d'entrada												 									14
	A.1	Entrad	a1										 									14
	A.2	Entrad	a2										 									14
	A.3	Entrad	а3										 									17
	A.4	Entrad	a4										 									21
	A.5	Entrad	a5										 									24

1 Introducció

En aquesta pràctica comparem diversos algorismes i implementacions dels mateixos per tal de comparar i experimentar amb les diferències entre els temps dels diferents mètodes, així com els avantatges i els inconvenients de cadascun d'ells. Per fer-ho, hem decidit implementar els següents mètodes.

2 Cerca binària

3 Hash amb taula

D'acord amb l'article sobre tècniques de *hashing* ^[1], s'han elaborat diversos algorismes segons les tècniques d'*open hashing* i de *separate chaining*.

3.1 Open hashing

Segons les indicacions de l'article anteriorment mencionat, en aquesta secció es troben les tècniques més eficients. Quan hi ha una col·lisió, la funció de *probing* és cridada fins que s'ha trobat la clau que es cercava o s'ha trobat una posició buida. S'han implementat les següents funcions.

3.1.1 Linear probing

Partim d'una funció de $hash\ h$ i d'un valor de passos i. Per tal de trobar la posició la qual pertoca la clau k, només cal aplicar la funció anterior h(k). En cas de col·lisió, sumarem tantes vegades el nombre de passos com siguin necessàries per tal de trobar una posició lliure. Així doncs, la funció resultant que haurem d'aplicar serà la següent.

$$h(k, i) = (h(k) + i) \mod m$$

Per posar d'exemple un cas específic, si *i* tingués com a valor 1, estaríem aplicant la funció de *hash* i, en cas de coincidir amb una posició ja ocupada, miraríem l'adjacent, i així fins que en trobéssim una de lliure.

3.1.2 Quadratic probing

En aquest cas, també tenim una funció de $hash\ h$ i un valor de passos i, però afegim dos valors nous c1 i c2. Altra vegada afegirem posteriorment i tantes vegades com siguin necessàries per trobar un espai que no estigui ocupat. La funció a aplicar serà la següent.

$$h(k, j) = (h(k) + c_1 i + c_2 i^2) \mod m$$

3.1.3 Double hashing

Introduïm una segona funció de $hash\ h_2$ i reanomenem a la primera h_1 . El que farem doncs és calcular el nombre de passos a partir d'una segona funció cosa que ens farà variar el valor segons la clau k que provem en cas de col·lisió.

$$h(k, j) = (h_1(k) + ih_2(k)) \mod m$$

3.1.4 Cuckoo hashing

Finalment trobem una versió una mica més complexa. Tot i que en l'article anteriorment mencionat utilitzaven una sola taula, nosaltres hem decidit implementar-ho amb dues. Així doncs, tenim dues funcions h_1 i h_2 . Procedim de la següent manera a l'hora de trobar la posició de k. Apliquem la funció h_1 , en cas que la posició a la primera taula estigui lliure, ja haurem acabat. Si la posició no està lliure, substituïm la clau que anomenarem k_2 per k. Ara, aplicarem la funció h_2 a k_2 i la col·locarem a la segona taula. Si la posició resultant està buida, ja haurem acabat, en cas contrari, aplicarem la funció h_1 a l'element k_3 que hi havia on ara posarem k_2 a la segona taula. Repetim aquest procediment fins que haquem col·locat totes les claus.

Aquest procediment pot ser perillós, i és que mitjançant aquesta tècnica podem trobar bucles. Per exemple, comencem substituint una clau n per una altra m i després de fer l substitucions, ens trobem que tornem a substituir la clau n per m en la mateixa taula. Per detectar això, podem utilitzar un comptador d'iteracions màximes o vigilar expressament que no es repeteixi el primer cas, que és com nosaltres ho hem implementat. Una vegada passa això i per tal de poder afegir l'element, només podem fer un rehash amb dues funcions noves. En el nostre cas però, no afegim la clau i ho indiquem. Una possible manera d'evitar això últim, seria utilitzar x funcions de hash i no només dues.

3.2 Separate chaining

Per aquesta part hem implementat dues tècniques molt similars entre elles d'acord amb Grill^[1].

3.2.1 Move to front

Utilitzarem una taula i una funció de $hash\ h$. Cada posició de la taula no contindrà només una clau, sinó que consistirà en una llista de claus. Cada vegada que volguem afegir-ne una, només caldrà aplicar la funció h i afegir-la a la llista.

Una de les parts interessants d'aquesta tècnica és que a l'hora de consultar una clau, aquesta es mou al davant de tot de la llista. És una bona manera d'accedir més ràpidament a claus que són utilitzades més freqüentment. Altres maneres serien mantenir la llista ordenada i inserir les claus mantenint aquest ordre, tot i que això només tindria sentit amb un nombre important de col·lisions i realitzant una cerca binària posteriorment, cosa que va en contra de la idea d'utilitzar funcions de *hash* amb taula.

3.2.2 Exact fit

Aquest cas és molt similar a l'anterior però no utilitza cap llista sinó un *array*. La idea és mantenir totes les claus amb col·lisió en una mateixa regió de la memòria. Així, quan s'hagi de recórrer aquest *array*, no anirem saltant per diferents posicions de memòria sinó que accedirem a la posició adjacent. Això ho podem assegurar donat que la funció per afegir un element a un vector en C++ ens assegura que si els vector creix més de la capacitat reservada, aquest es copia en una altra regió de la memòria mantenint tots els seus elements en aquesta mateixa regió, i a més, adjacents evitant l'accés aleatori i realitzant-lo de manera seqüencial.

4 Filtre de Bloom

5 Experiments

Per a realitzar els experiments s'ha utilitzat diversos jocs de proves que podeu consultar a l'annex A al final d'aquesta documentació. S'indicarà per a cada prova quin número s'ha utilitzat. Les taules detallades dels resultats obtinguts es poden trobar a l'annex ??.

S'han realitzat experiments de comparació de salts (*miss* de la clau en la posició calculada), temps d'inserció i temps de cerca, diferenciant també entre els temps dels elements trobats i els no trobats donat que ho hem cregut adient i interessant.

Així doncs, a continuació es detallen els experiments realitzats separats per categories. Una vegada realitzats, s'ha seleccionat el millor de cadascuna i s'han fet les comparacions adients entre categories.

Per tal de procedir amb l'experimentació hem establert in seguit de variables que no es canviaran durant el procés. Una d'elles és la mida de la taula, fixada en 1,5 vegades la mida del diccionari d'entrada, així com el valor de passos fixat en 13 i els valors de c1 i c2 fixats en 11 i 17.

5.1 Cerca binària

Per omplir amb text...

5.2 Hash amb taula

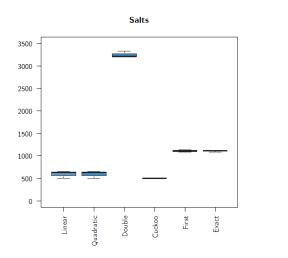
Per tal de comparar els algorismes de *hash* amb taula implementats, hem realitzat 5 execucions diferents per a cada joc de proves i tècnica. A continuació es detallen els resultats obtinguts amb la mitjana dels conjunts de dades 2, 3 i 4 amb mida de diccionari 10.000, mida de text 20.000 i d'un 20% aproximadament de la mida del conjunt unió entre els dos respecte al diccionari.

En el primer gràfic de la figura 1 podem observar com la tècnica que genera menys salts de totes és el Cuckoo, seguida de ben a prop pel *linear* i el *quadratic* que tenen pràcticament el mateix valor. Veiem que les dues tècniques de *chaining* han obts una mica tingut uns resultapitjors a les d'*open hashing* generalment.

De cara a experimentar amb temps d'execució, cal recordar que s'ha realitzat una mitjana de cinc execucions per a cada joc de proves separat per obtenir uns resultats més fiables.

Pel que fa als temps d'inserció, podem veure clarament com tot i que abans el *quadratic* tingués el mateix nombre de salts que el *linear*, té un temps d'inserció lleugerament inferior, tot i que això podria ser degut a la *caché* de la màquina i per tant, tampoc podem assegurar que sigui així. Si mirem Cuckoo, veurem que efectivament té un temps inferior d'inserció. Això ja era d'esperar veient el menor nombre de salts que feia mirant el gràfic anterior. És així donat que generalment una col·lisió la pot solucionar només amb un salt donat l'ús d'una segona taula de *hash*.

Finalment, veiem com les dues tècniques de *separate chaining* tenen un cost superior. En el cas de la tècnica amb llista (*move front*), veiem que el temps és inferior al del vector (*exact fit*), cosa que és degut a la relocalització d'aquest últim quan les col·lisions superen la seva capacitat, i per tant, tenim el temps afegit de moure'l sencer a un altre lloc de la memòria. Cal recordar també que el cas del Cuckoo no contempla el *rehash*, cosa que provocaria un augment considerable en el temps d'inserció total.



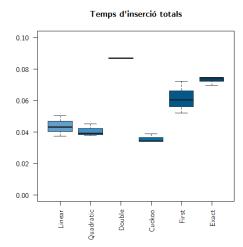


Figura 1: D'esquerra a dreta, la distribució del nombre de salts i el temps d'inserció total.

Pel que fa als temps de cerca d'elements al diccionari, i com s'ha comentat anteriorment, hem separat els temps de cerca dels elements que en formen part i dels que no.

Veient els temps de cerca podem trobar dues regions molt definides, tal i com es veu en el primer gràfic de la figura 2. Les tres primeres tècniques d'open hashing provoquen un augment molt important en el temps

de cerca total. Aquesta diferència es veu molt clarament que prové dels elements que no formen part del diccionari tal i com es veu al gràfic de la dreta de la mateixa figura. Això és així donat que en aquests tres primers casos, anem fent salts per la taula buscant l'element indicat, així doncs, ens recorrem la taula (sencera o no) fins a trobar un cicle, és a dir, que tornem a mirar la posició que hem mirat la primera vegada. Com que l'element que cerquem no és a la taula, el temps de fer la cerca serà el del cost pitjor, justament per aquest recorregut.

El fet anterior canvia però en el cas de Cuckoo i de *separate chaining*. En el primer cas, només hem de comprovar dues posicions, així doncs, el cas pitjor només comprendrà un sol salt. Pel que fa al *move front* i *exact fit*, si la posició on hem anat a parar després d'aplicar la funció de *hash* és buida, no farem cap salt donat que no hi és, si té algun continut, només haurem de recórrer la llista o vector que tot i que podria contenir tots els elements, no passa en el cas mitjà (tenint una funció de *hash* mínimament correcta, un mòdul per exemple, ja ho evitem en la majoria de casos).

Per acabar, si ens fixem en els temps de cerca d'elements compresos al diccionari, veurem que efectivament i tal com hem dit abans, el temps de cerca d'elements a Cuckoo és inferior que als de les tècniques de separate chaining donat que farem com a molt un salt, és a dir, que limitem a que la llista o vector (per fer una comparació amb els altres dos) tingui mida 2. Recordem però que el temps d'inserció total és més baix degut a la no implementació del *rehash*.

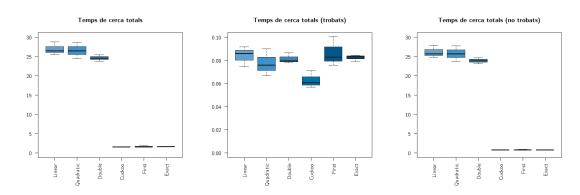


Figura 2: D'esquerra a dreta, la distribució del temps de cerca total, cerca d'elements que pertanyen al diccionari i cerca d'elements que no hi pertanyen.

A part, també podem veure l'evolució dels salts i dels errors en l'inserció per cada joc de proves. Ho visualizem molt bé als gràfics de la figura 3. En l'eix inferior de cada gràfic hi ha la indicació de quina entrada correspon a cada valor. Donat que les entrades 2, 3 i 4 tenen la mateixa mida, s'ha triat la 3 donat que s'aproxima molt a la mitjana de les tres.

La gràfica però no ens ha de fer veure que el nombre de salts és exponencial en relació a la mida del text. Cal tenir en compte que la mida de l'entrada 1 és de 16, la de l'entrada 3 de 2.000 i la de l'entrada 5 de 20.000. Observem però que en el cas del gràfic d'errors d'inserció, només n'hi ha un que augmenta mentre que la resta romanen a 0. Aquests valors corresponen a Cuckoo i es produeixen quan troba un cicle.

Pel que fa al gràfic de l'esquerra, de baix a dalt de l'entrada 5, les dades corresponen a *linear*, *quadratic*, Cuckoo, *move first*, *exact fit* i *double*.

Si ens fixem però en els mateixos valors però relatius a la mida d'entrada i tenint en compte tal i com s'ha dit anteriorment que la mida del conjunt unió entre el diccionari i el text és d'un 20% aproximadament respecte al segon, podem veure que efectivament no hi ha cap creixement lineal. Tot al contrari, podem observar que a mesura que l'entrada creix, el nombre de salts i d'errors es redueix. Segurament s'estabilitzarà i ho entenem com una manera d'aproximar un percentatge de salts i d'errors per qualsevol entrada aleatòria.

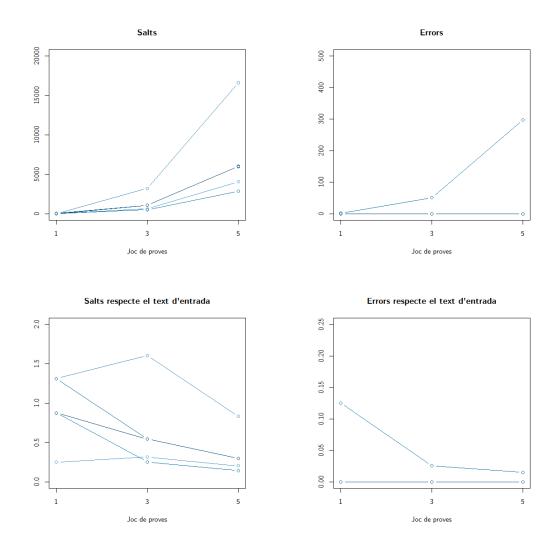


Figura 3: D'esquerra a dreta, l'evolució del nombre de salts i del nombre d'errors en l'inserció.

5.3 Filtre de Bloom

Bibliografia

[1] Grill, Bernhard. A Survey on Efficient Hashing Techniques in Software Configuration Management. Vienna, Austria: Technische Universität Wien, 2014.

A Dades d'entrada

Les entrades estan ordenades de menor a major mida. L'entrada 1 té una mida reduïda per proves ràpides, mentre que la 5 consisteix en un text de 10.000 caràcters. Pel que fa a l'entrada 2, 3 i 4 tenen totes la mateixa mida de text de 2.000 caràcters i són les utilitzades principalment pels experiments. Els nombres de totes les entrades han estat generats aleatòriament assegurant un conjunt mínim resultant de la unió del diccionari amb el text en cada cas.

A.1 Entrada 1

Diccionari

R

9611 7255 5176 3196 6664 9744 9540 820

Text

517682 7255 5176 3196 6664 9744 9540 820 4937 9195 1875 1686 1091 1084 2376 3021 3458 6777 2547 4569 430 8285 1265 887 912 9856 2073 870 2025 8525 2313 2024 3790 5383 5626 5451 134 7865 6511 2322 5250 2060 6211 7730 5358 9611 7255 5176 3196 6664 9744 9540 820 4937 9195 1875 1686 1091 1084 2376 3021 3458 6777 2547 4569 430 8285 1265 887 912 9856 2073 870 2025 8525 2313 2024 3790 5383 5626 5451 134 7865 6511 2322 5250 2060 6211 7730 22

A.2 Entrada 2

Diccionari

1000

1913 1921 3629 1449 780 4660 4335 8781 1303 9840 1317 3001 4557 79 4413 2983 790 7678 4660 3686 8779 6562 3184 8580 9620 2814 1240 1272 9134 2688 3142 908 2844 6816 5749 8190 5876 8680 9140 310 4672 4032 7670 7955 6560 6581 6190 777 6421 9486 5132 7226 1835 892 5561 6149 200 8152 3326 9546 783 773 8853 5710 635 7080 3140 3205 9962 8763 215 3868 3766 576 5257 2679 6069 7588 8123 1105 6253 3310 4263 9409 3856 5316 334 4766 806 6007 2111 503 9389 8357 1253 5874 196 2762 5816 9197 7738 3789 4793 3814 6362 2675 2478 1371 4401 6000 6237 5981 2238 1404 5148 1934 665 8474 6087 3474 5924 3857 5743 4014 1350 4790 9622 2561 2810 906 9616 2271 1277 1698 1727 7034 3103 2455 5708 5940 8301 7600 3338 8626 5884 4084 3313 3660 3156 5101 1103 6899 7255 1646 8960 9658 2449 5542 4272 7560 6878 9725 8312 2712 2083 5899 6080 6369 860 7198 9307 3941 19 1467 9606 9652 9742 9158 8873 2365 3527 9444 4109 3314 3268 2807 8929 5620 3353 5461 9700 314 8842 6909 9223 9263 3031 8195 8291 8134 393 5028 2554 9734 6012 3244 2535 4694 1607 2844 1683 7554 3712 2683 1739 2340 5291 2273 5570 8405 5193 9126 6251 6687 7088 273 9759 8578 8524 2596 8 4762 5325 9924 6104 6742 5415 5747 2252 6048 1348 8787 4021 2196 2753 8341 8352 7647 5618 3490 7237 1657 6512 9014 9535 56 6805 3052 1724 1566 5 5735 734 8100 7736 7459 403 628 4751 3297 6658 4485 5841 4786 3276 3861 7309 7728 8229 9094 9478 3834 4770 4445 1316 2945 7729 3888 5173 5359 3693 3404 9843 6521 4645 8511 8388 3504 2055 6647 3093 3887 3815 7893 4213 660 8223 2277 4053 5578 718 1272 9674 8361 9611 3106 3636 6157 2787 8041 9449 9436 3292 7366 8879 6136 4269 3655 3142 629 3845 999 1865 2796 6817 5645 2847 7054 5599 4428 7568 3870 183 4952 7037 5139 9793 6766 1744 7169 2986 4080 2608 9123 8988 8201 1564 2226 8954 4645 2527 6198 197 9300 3780 6642 3622 4807 3559 7819 567 4227 504 6457 7432 4524 8990 3529 1212 7565 6044 7907 2863 9402 6840 1341 7822 9630 606 7565 775 5575 9271 7619 4486 3548 8098 291 7307 3941 8330 5527 4349 2594 5892 7661 5732 4121 2002 6043 4440 62 3406 6992 9188 2836 1792 8929 598 9250 6999 425 3043 4039 4886

8019 5611 4253 5829 771 8698 6369 6773 4162 8229 506 5395 5892 7540 960 1963 364 5044 203 5507 7405 1287 3828 7981 5695 3484 8987 9584 9655 6345 5207 7686 9077 8949 6363 4141 5326 2399 1122 6498 733 8120 7912 5559 6944 1211 5210 8343 7671 2814 930 5416 4750 2113 2605 9081 721 4176 9417 4023 7193 4017 2952 3677 9678 6722 9822 9833 8935 941 1216 5308 6412 9515 3150 8884 3187 5307 3946 1044 1611 7549 849 486 8765 4172 6946 6640 3017 5133 7101 1980 2868 9520 4198 8634 2100 236 1959 7514 8472 3307 3464 448 1948 1545 7983 4827 4260 1649 9401 8550 887 9908 8773 2972 9311 6907 5886 4833 466 6054 2662 6098 6405 2921 8275 8366 7924 4817 2398 9336 2635 9859 3498 7634 495 1897 629 8070 2297 3479 3929 3825 8928 2422 9808 9416 1398 1874 9607 5001 3451 1093 104 6479 2591 4805 227 7943 7739 5701 185 9365 7858 6843 8674 9645 3938 3282 6396 9048 5083 6159 4880 7635 7574 4821 9942 899 1682 2799 2846 1794 6987 161 1885 8751 9032 4231 4320 4050 4074 5501 4692 6409 1719 288 9867 7067 6547 9941 9999 2025 8472 5211 4610 6229 8486 3841 3421 1191 1342 963 6126 6217 3383 5946 3898 4224 4077 4904 8836 5999 1951 9216 6503 6769 6435 7299 8926 605 1602 359 2522 5783 2010 6315 2913 203 9995 8520 9616 283 4650 252 7801 7369 5609 6930 9636 9002 1049 9186 3450 6312 1116 3516 97 5888 4678 8216 3305 5120 6190 484 5850 8564 8802 7083 6414 1421 5151 7106 5902 9813 8205 2557 976 1321 7000 5401 8965 5867 7155 627 2505 9544 2955 3546 2437 8984 2647 9749 9334 2781 5677 4464 6114 3752 7973 2269 4520 2335 3563 4175 4274 7990 7907 1188 8786 5654 6898 2576 7150 6187 4064 5483 4836 2662 8674 9377 7116 4442 7459 4148 2450 5964 8321 5909 6998 3208 7657 7378 4669 4166 993 5707 5373 7006 6810 9802 5890 8032 5990 1312 191 9228 4409 9197 3954 8567 869 9362 650 683 6792 4872 9383 2104 5214 6622 7100 1280 2534 8964 9427 7376 7771 845 8182 6724 7203 2376 7123 6054 7488 699 7568 5871 5263 2860 7497 8588 8245 2938 6685 9211 4402 5131 4537 5222 1807 332 7962 2719 5250 7648 6360 5722 4090 4641 459 9299 7731 4381 8641 8617 5052 7920 478 9896 1950 1792 4665 9363 8894 2196 6112 8361 7915 8563 6902 7803 6611 6228 1202 7262 3944 5486 1636 4753 6959 4614 68 1827 621 6461 5155 143 830 7230 6499 4314 1975 8538 6026 6273 9359 6818 494 1604 4503 5841 1582 203 9954 7503 4130 6152 4996 829 4377 6871 512 8595 5935 2388 8466 7924 3440 3102 873 8535 4139 1464 7230 2470 5305 3012 501 3278 121 5937 3687 8917 7395 3393 6931 4557 1875 8010 9052 1367 165 7643 2188 1682 578 2027 9374 9231 3179 5496 6647 736 5923 3553 3211 5102 6600 9752 1169 8653 1611 707 591 7503 7652 4772 1120 5892 3327 2537 2479 4799 6662 9101 4699 5366 7230 1642 6635 5043 6078 3849 4733 7654 9817 937 4520 8087 4037 227 9296 8512 1292 1752 7409 5441 4391 3316 983 8970 7389 7154 7667 3225 3093 7732 5939 4395 1367 2492 6878 3477 8774 3463 6072 4411 2459 5709 9090 6863 5398

Text 7381 5829 333 2388 5446 7079 8597 881 6031 9677 3195 5419 4204 3725 7331 2548 8590 305 2508 8250 640 9719 3244 611 7707 1557 1480 4494 2259 1311 3455 2093 7099 4543 2161 5201 9577 327 9421 2302 7511 2008 628 1778 1453 2065 3268 9144 1286 3612 7560 8345 6593 4757 4450 4673 9238 7696 2054 4088 1627 2551 7872 9847 6867 5884 9169 9368 8795 5459 3397 6607 8677 3304 7090 7031 4016 3698 8002 5104 6240 2150 8568 8538 1726 2983 1508 4099 505 3372 5121 3813 4262 4536 1146 4242 8884 7551 3767 1180 9096 1158 7389 93 6565 8115 2723 6705 9525 7874 7526 6789 5715 2932 9311 5223 2613 3639 919 278 6248 4452 4121 694 160 1870 8955 2592 2678 2437 7301 8286 7381 5040 6928 9100 298 2859 5765 5022 7727 6040 1705 5978 327 2539 1190 3619 4877 5454 8815 8097 6988 6341 9072 2256 6180 4907 3105 8751 9169 2383 3706 4418 1444 1256 2577 4645 952 8633 5429 8867 1737 136 5542 8887 9385 5942 6288 3898 4314 3023 4631 3559 489 1824 4332 2682 1170 7330 4355 4939 5474 2361 3108 6229 7856 2076 8651 1198 1193 7040 5561 2372 3267 9883 1722 973 3250 5958 3953 7645 8061 1221 2130 901 9784 3419 8229 936 2078 5823 4966 466 3867 3153 2246 5418 2450 4833 1634 7924 8375 568 3950 3304 6620 2689 8303 7814 4849 1144 33 3772 1829 2963 9391 5063 7898 2164 2675 8423 8968 2620 8702 9538 4579 4559 766 4986 2248 6741 4881 5182 19 9936 816 7453 7306 9816 5487 1646 8269 9407 8079 5468 6926 1448 5874 5640 1734 3230 9375 9547 3541 6901 8592 2616 7701 5954 1032 4560 5615 3510 6483 9382 8725 578 1738 9523 8544 4161 3144 5971 1867 8670 142 2500 9791 9610 8711 7685 2051 6424 125 9066 912 4227 9950 5398 6770 1424 5647 3553 347 3453 6897 948 5955 1591 6399 6318 7049 4140 9682 1208 4102 7062 488 7114 2550 1554 8489 2241

A.3 Entrada 3

Diccionari

1000

7763 4623 4197 2443 2748 999 1828 1168 3141 3128 8264 4868 7454 7209 6136 9132 2370 6669 5068 7652 5997 2331 7097 5791 7693 9186 2543 4449 7969 3328 8265 1717 5064 2843 5167 2463 8149 5336 6390 9383 471 2432 3085 4569 666 3259 3134 3514 3405 7778 3240 1961 2583 5338 8698 713 3837 3242 9825 4488 9633 6590 4619 6611 9498 5895 8619 6859 4593 123 5563 5325 2448 8743 3036 3439 7517 907 1302 5036 1822 7223 276 5284 11 6732 5411 1240 5249 4172 2657 2200 236 1963 467 4606

3500 5646 847 7648 932 4653 9144 7239 2687 6577 7585 4359 7712 9763 5184 1498 8776 1223 8228 2076 1339 4596 6781 7462 9483 4479 3137 8048 6601 1429 2147 9781 5807 2311 5913 920 730 504 1486 8076 4225 423 2856 6840 136 5151 1969 1305 1774 959 4587 870 8911 4812 8618 6490 6699 7911 7871 671 7500 2262 6414 8475 4509 8711 5263 4565 6719 8662 7482 7149 7147 8739 1444 5194 6548 5445 928 8416 4902 7385 3685 3922 7017 2389 5900 7821 3929 8106 8150 448 5898 8135 5480 5008 2547 8725 3509 9948 8153 9525 9965 112 6051 5128 165 2650 5483 2658 9934 9959 2045 7418 7053 396 1060 7818 7334 3932 1232 3990

A.4 Entrada 4

Diccionari

1000

107 9669 4014 3809 4119 9488 1789 1515 676 4245 1807 6877 2720 3314 1092 5769 4249 6383 1124 8150 7832 3162 3440 8861 6951 8197 987 1012 7678 2642 4986 5648 1421 4349 1366 7310 1635 252 6892 8387 8817 355 489 6315 4869 4865 1492 1145 4015 7144 6456 8990 3576 7774 4083 6259 2398 5273 7559 7008 1982 5319 2911 952 2801 4975 955 5192 5425 9620 9586 7898 9264 6584 4592 6338 8237 1212 7336 61 6378 3768 3760 769 6454 4696 222 6786 5930 7094 7985 6966 6967 194 6416 1328 4063 9956 6366 6270 5825 73 3709 8491 4354 9820 2827 7360 1864 1880 1974 2755 8261 5150 972 5732 2032 5075 5451 9758 3281 6853 7771 778 8257 5780 8835 6625 4486 4322 3636 3645 4521 9156 6623 2936 3619 276 5348 9071 8247 7281 1592 2290 4433 5968 8489 9844 6967 5798 7597 6195 968 1881 5746 6763 6771 9866 3483 5527 5765 9419 3960 8255 7696 7106 3860 7427 3087 3713 4554 1561 2412 4430 2756 3953 3413 2392 7378 9123 6206 9312 9945 6120 1785 7079 2816 8530 5345 4824 7550 6459 4839 7063 1711 2353 6002 3112 1894 2823 2452 2354 2763 4195 1243 2810 960 6906 5580 8337 7079 4214 7545 4172 4694 3293 5628 3962 5874 147 9899 6605 625 4587 4903 5277 6887 3493 1129 8017 1234 7737 5678 4796 9854 7456 6965 810 5830 5371 2585 8571 2928 3423 9197 7510 231 5825 2697 9068 5091 2286 5145 877 9385 1863 2773 5079 4219 7856 5646 9396 5806 4132 2619 2117 6262 9638 1580 5613 9419 4610 4092 4378 2474 4739 3085 4712 1758 3375 3436 7887 1713 5932 959 5832 7941 2358 2483 2253 8381 6962 8038 9922 4688 5108 8837 358 2305 9419 9283 2713 1168 7902 5457 1384 8311 3908 3456 919 1235 821

Text 1855 3177 4852 5860 8754 3960 7266 8053 677 4992 8491 1839 8330 5354 1315 9698 9592 5781 8328 6827 7028 98 5825 5202 3638 9016 8017 5516 3211 4681 9600 6108 7280 6825 547 5948 64 1215 9844 784 7386 1281 7588 1701 795 378 2690 8038 537 4602 7580 1222 6059 6955 562 7025 9392 2098 1711 4360 7641 6069 5552 8330 7886 4218 5482 9023 4303 2345 7712 4505 9625 1686 5396 5476 9064 2343 3127 8814 1690 8663 8576 1320 6087 1707 4954 5484 674 2188 5596 5404 7187 3321 7231 3348 1752 1345 1861 3970 1811 2202 167 2061 9590 7316 1841 1328 8848 8278 6385 2690 948 4490 5955 2591 2506 5092 2905 1702 2403 1010 2996 5943 7129 4324 5615 9612 7163 7778 2684 6254 4616 9617 5071 1385 870 1988 3655 9305 8491 8139 918 8822 9244 8407 1114 9885 9962 649 2698 7677 8045 254 1826 414 9354 2372 6608 7181 4671 8879 7220 2083 7584 3396 7793 9044 919 78 2152 2233 3355 8994 6208 8614 9376 4785 5460 7686 9289 8423 8094 4020 7615 8297 9782 6737 540 6628 1711 5654 2941 407 8021 7984 1629 6464 5181 5450 9487 1526 9095 8828 8606 56 1226 7623 2211 1837 562 4916 482 9254 9672 1930 4995 6445 1 4599 9351 9148 6898 4442 5586 2329 2710 5017 5478 7059 7836 5206 8951 4328 8254 1069 4579 7642 2304 6017 5610 1249 8580 2646 957 6283 476 4006 8996 2012 4166 9517 5896 2361 5365 3749 653 8760 2056 6616 1586 1599 2303 1754 1075 4351 2697 6044 5313 7137 3249 9614 9970 2314 8410 5770 9035 2038 4711 8697 1489 4995 4455 4476 7659 2155 2140 8712 1831 6732 8331 2436 4569 2277 7245 8575 6105 2019 8253 5858 5559 8797 8602 1969 9917 7052 1917 2656 370 580 1122 4989 1654 6484 8415 6568 299 1864 5453 9671 2004 6324 7723 83 7198 5001 5985 7858 4481 6929 8010 4231 5675 7169 9071 3127 1342 5272 3044 3432 8934 8967 3968 7703 1658 5271 7407 9665 4003 8376 3809 8795 8526 6781 1238 1657 2228 9505 3491 2219 6504 1114 4223 2592 7228 1930 3220 4246 5635 2989 2896 4617 9670 2557 3286 1945 6683 4971 5451 3224 4845 1584 5272 6440 7982 5154 4455 4511 7187 8303 4 3063 1394 2639 957 8499 3262 1904 7325 2423 9606 8336 5908 5439 5727 224 1543 6398 2586 8407 9165 11 6768 1287 3475 9628 7381 538 8079 2389 8073 8442 3522 8594 8357 9068 3082 4066 499 7657 3079 5124 6654 7562 6818 6608 6448 4022 7743 6398 4191 8458 5592 4542 1565 8573 274 5106 2377 5657 8915 2186 3707 1518 1570 6906 4193 8271 7020 1972 3363 2466 4168 2987 9511 4040 4151 7593 7759 5105 7947 9721 7800 4134 9160 5446 9271 4309 1840 2193 1070 6541 8412 9098 8435 4507 7660 5017 7331 6946 8494 9177 4140 6623 4786 8072 3953 7208 3407 8205 3062 6253 2410 3506 1543 9041 6029 5704 2920 2404 6103 3655 8537 1989 8440 6930 6814 931 2237 7196 3708 6567 3938 4663 9052 4086 5890 2721 5961 7345 8343 3865 177 8411 9617 1926 8984 2207 7409 7664 4845 2045 9250 2707 9450 2568 246 366 9384 5460 7869 652 8454 8567 1169 5394 7018 2617 1622 4856 8728 701 4649 5504 47 1210 896 1873 2670 2596 5862 2164 2287 541 7779 718 2177 3440 6808 7118 8946 7784 532 4859 6433 6339 6509 8190 3377 6460 2211 9342 1927 6041 2882 8837 1276

5718 3756 5522 5971 2403 9676 4792 9100 7113 8473 4313 9830 2810 4193 601 2053 8300 7169 490 5192 6328 5376 5535 9220 1263 2986 5407 9265 3762 7495 40 4294 6865 2332 2433 8617 7231 8016 114 4737 9990 7057 4844 6074 1347 9258 5436 2495 2774 4174 6560 698 4606 4936 1676 6192 1638 1441 1219 1606 4460 5242 8970 3012 9700 5475 5247 8399 1992 3339 213 9844 3299 5051 2962 8781 6618 2481 883 1959 7298 9088 5890 7802 4596 9773 9923 2782 1387 2062 2703 980 1468 9459 9894 7879 337 70 5223 3966 3140 279 7753 2020 4680 1425 1395 1554 4426 4785 587 5325 7293 897 4782 7868 8286 1542 9489 4941 6742 7981 9373 5562 3559 5082 8061 3909 577 6878 9997 3326 8320 236 5577 1113 1201 2806 1902 2452 2592 9827 8694 16 2057 2633 1148 2647 8964 1523 7270 4578 7513 2651 2339 8168 6506 2363 2810 6180 8308 7075 4473 270 354 5959 9292 5886 8773 4210 512 3956 5553 6845 2418 4844 4738 4427 5019 3184 8265 4875 6812 7098 2644 5292 8069 4632 5023 5005 8157 7254 2075 8792 51 710 5340 6804 3704 3581 2633 6284 8953 7787 6132 7669 7947 2358 5537 1155 7932 4644 2496 5176 1586 9657 8680 9054 8028 4030 4318 9329 7377 866 7903 8387 7876 6631 1389 727 2938 8117 1772 503 6799 4707 8873 1005 6880 8051 2238 9554 4493 7959 2430 158 8380 5884 4804 9155 1490 9983 9612 6626 9558 5935 2 7306 3670 3515 2363 6821 211 1888 7970 4303 9604 2371 2738 4941 5306 1526 464 4308 4573 6115 3746 3944 4893 4362 3052 2989 3829 8041 4461 257 1097 4896 8816 4545 7820 3231 829 6283 2514 4678 8753 4832 8246 2820 6132 200 7952 4855 5272 425 9998 4660 8720 5011 9719 5689 5341 1281 1945 4200 18 1481 4687 7480 7573 7577 7014 4075 3137 3799 9490 2353 885 7110 3468 2012 6379 5366 9732 9069 4371 5897 8632 8352 6004 662 503 2912 5690 9664 9999 5011 2438 5658 1845 9672 5231 8620 1907 4140 5500 5639 9405 7934 8771 2986 8036 8745 4676 8293 9555 9013 715 5326 6906 8036 8628 5600 4094 782 8496 2989 2765 8457 2204 2534 2550 2573 6061 1686 7117 8816 4630 1513 9021 9090 4495 2288 7444 5604 248 8739 4031 3786 8757 6276 9128 3240 5262 2264 9114 1916 1066 1958 7664 146 3853 6566 9094 4173 2525 1506 4338 9245 8731 5971 6416 980 5338 867 913 3859 1052 2120 5112 4456 2307 7245 7739 6213 7012 2098 8317 5776 3597 6665 6004 9844 6390 1864 9379 1170 2092 838 8947 1016 1064 8925 9867 5643 3522 2805 4680 5276 750

A.5 Entrada 5

Diccionari

5000

1267 41475 43251 38732 87693 84266 54309 44815 41260 28364 39271 36718 61546 83712 60175 90490 89645 19362 68988 16853 14520 90784 13464 67477 454 60152 67319 91118 12553 50912 67315 60935 71624 43219 70999 31639 61242 64984 5022 15925 88959 90176 44350 30887 18488 23687 48429 34986 38059 20781 7613 8891 95314 7960 93716 66862 82792 74310 14492 60129 82430 95858 82478 47196 37144 92263 97773 49759 25277 77384 48440 40807 52282 52234 83750 29612 25205 8501 55568 987 35084 65835 77903 93310 10555 24166 53934 65594 12487 81877 16659 91116 25321 388 78031 28175 43563 6657 61892 55296 2885 20103 17099 45483 8971 53881 67399 62403 82791 64155 79088 60816 74541 89858 12614 51704 64112 65776 71037 610 30133 80217 33739 78665 92621 62471 50936 4319 84861 53988 769 32034 57431 97600 89701 82484 48920 17453 79236 33409 51313 33143 6818 9400 68177 13994 75906 37742 29047 53077 58127 38179 93116 10245 2394 14941 46165 58304 19703 10745 35774 79550 99801 34179 23884 33433 4700 22022 49058 61631 95531 72178 19817 51939 95308 89725 34298 8789 66223 37944 42858 35098 62844 39325 38140 6147 29341 10086 94847 45693 61671 10143 18630 37380 97893 53814 93598 8806 92538 11700 23604 87772 84843 42801 33302 26175 43307 66681 88137 48274 75947 98777 42586 45364 53273 55469 54416 1848 65873 42651 13538 6568 39135 87662 71585 40531 9765 89128 62679 87105 16839 97428 12641 80847 14425 21477 21243 9686 84498 64848 92640 57968 98012 12531 2088 7183 10516 69111 76523 51570 94256 62174 61406 71593 69829 52162 1983 32432 11093 89276 67377 94783 89167 87168 680 78080 80838 24531 49964 43477 2446 26696 38233 43868 98151 56131 40874 30249 62609 17426 11991

75780 78516 79373 9624 90300 94208 70111 90430 27370 9968 73584 83785 36790 64742 58455 23871 95889 25958 20457 20704 34074 58255 10001 70433 34487 9969 87535 18628 22176 98355 46471 97544 91154 22926 69640 70269 69780 91857 44390 30211 28533 72836 13508 92759 92244 80780 2390 92092 32674 79731 87718 74961 85472 36800 80170 98371 8451 93270 96652 42677 355 22064 19844 28580 84957 21214 50807 62555 97157 82375 32493 19304 40393 71776 32242 36854 49368 43493 13687 28409 55425 23793 6636 98213 68699 93715 41267 292 73090 90837 26075 37137 54219 57494 46967 39318 81774 29672 72942 76454 44259 31806 7755 81261 27519 8751 21106 32227 86762 96944 41687 96517 35274 98434 1244 23789 252 72762 33565 53254 72296 59902 20258 15214 69139 18884 58084 96421 1397 69150 83269 20607 80899 30013 62195 5429 45884 37970 24199 8603 65638 22441 20787 77177 67955 79002

Text

27497 49818 38085 34928 17640 15771 46059 13278 38939 11088 36796 50448 64404 50344 25263 57636 20614 15063 77813 42409 58887 13498 98050 45516 34674 89032 94555 5283 97954 53269 7286 53629 18642 51172 27989 92061 56871 20819 76671 48734 80681 99195 90516 22480 68509 92970 435 58493 37332 35009 34322 98394 85522 23347 78425 1488 70845 50448 78897 25839 63873 24250 76789 1892 43129 60479 10055 77944 70871 84182 30602 62267 1480 78660 11746 12271 17388 56891 28262 42229 13075 30392 41490 35139 67964 71222 43530 13279 139 12053 31348 62594 80126 96301 13077 55918 20962 81212 96113 66530 72337 33840 77377 31482 8317 55780 84064 47616 4020 18565 47788 48322 70582 87282 44413 33096 84364 98338 19122 34010 54951 23477 97616 73182 26663 65080 63968 97738 86825 23002 95778 54854 46450 64727 89160 2474 11178 67614 70653 36599 9450 16777 21463 90526 171 72927 85591 12780 16554 55208 90965 31429 68860 13792 77311 21195 27895 21901 53931 81221 54855 79449 1167 36170 82936 33436 95675 53335 26668 40927 55063 45659 57625 92942 26131 6531 28604 89472 23116 64166 72887 12923 87176 89490 77273 22042 16026 95648 92922 10769 82799 63668 81760 54804 75142 77811 77333 18100 97089 50739 46374 21795 88999 75773 21093 16034 7665 62937 93188 27024 69216 71651 57696 69286 62410 44212 78704 77272 54133 29745 22345 60682 95822 97599 34458 45026 66100 68796 31260 61177 16385 35614 53198 80846 42309 58692 66109 40796 12524 51093 29841 19416 3581 48730 85080 19289 2851 14651 19757 89595 69086 2512 7927 29379 99112 89600 44510 66387 37684 33572 46732 90292 30184 68358 97940 85909 57190 6761 54541 96542 47525 65376 27691 7924 39405 12861 84311 97816 60666 93439 29258 39931 96131 44942 87503 58065