Tabele de dispersie

Ne punem problema menţinerii unui set finit de date, fiecare dintre ele identificată printr-o cheie unică, şi notăm cu K mulţimea cheilor din acest set, numită *mulţimea cheilor actuale*. Prin natura lor, fie că sunt întregi de lungime fixă, dar mare, fie că sunt stringuri de caractere, cheile actuale K fac parte dintr-o mulţime U, de cardinalitate mult mai mare decât K, numită *universul cheilor*, lucru pe care îl notăm $|K| << |U|$. Pe această mulţime dorim să facem operaţii de inserare şi ştergere, şi, de asemenea, foarte frecvent, operaţii de căutare. Ideal ar fi ca operaţia de căutare să aibă timp $O(1)$ sau apropiat de el, în orice caz timp constant (şi mic) în raport cu cheile din U. Structura de date care ne permite accesul în timp $O(1)$ la orice componentă este vectorul. Dacă ne-am putea permite menţinerea în memorie a unui vector T a cărui mulţime de indici să fie chiar U, atunci pentru orice cheie actuală $k \in K$, în locaţia $T[k]$ am putea menţine data identificată de cheia k. Un asemenea T se numeşte **tabel cu adresare directă**. Din păcate, pentru multe probleme concrete nu ne putem permite această soluţie deoarece ar conduce la o risipă de spaţiu. Dăm mai jos câteva exemple.

1. Vrem să menţinem înregistrări cu date despre cei 68 de angajaţi ai unei firme. Fiecare angajat este identificat cu ajutorul unui număr de cod compus din 4 cifre. Cardinalul mulţimii cheilor actuale este $|K| = 68$, dar universul cheilor U este format din mulţimea tuturor întregilor cu 4 cifre în scriere zecimală, deci $|U| = 10^4 = 10000$. Un tabel cu adresare directă ar însemna să menţinem în memorie un vector cu 10000 de componente, dintre care doar 68 ar fi efectiv folosite.

2. Vrem să menţinem înregistrări cu date despre populaţia României. Fiecare locuitor este identificat cu ajutorul unor chei unice, codul numeric personal, un întreg compus din 12 cifre. Cardinalul mulţimii cheilor actuale este aproximativ 22 de milioane, $|K| = 22 \cdot 10^6$. Universul cheilor va fi mulţimea tuturor întregilor de 12 cifre, deci va avea dimensiunea $|U| = 10^{12}$.

3. Anuarul telefonic al unei localităţi menţine informaţii despre abonaţii identificaţi cu ajutorul numelui şi prenumelui, deci un şir de maximum 20 de caractere deci va avea dimensiunea $|U| = 26^{20}$.

În toate situaţiile descrise mai sus este nerealistă folosirea unui tabel cu adresare directă, adică a unui tabel T indexat chiar după U. Trebuie să găsim o mulţime de indici pentru T, de cardinal m, mult mai mic decât $|U|$, eventual apropiat de cardinalul lui K, şi o metodă de a dispersa cheile din K pe mulţimea de indici $\{0, 1, ..., m-1\}$. Cu alte cuvinte, avem nevoie de o funcţie, definită pe universul cheilor şi cu valori în mulţimea indicilor T:

$$h : U \to \{0, 1, ..., m-1\}$$

cu ajutorul căreia să găsim locul fiecărei date din mulţimea noastră: data identificată printr-o cheie k, $k \in K$, să se afle în locaţia $T[h(k)]$ a tabelului T.

O asemenea funcţie se numeşte **funcţie de dispersie**. Pentru a fi o funcţie bună de dispersie, funcţia h trebuie să îndeplinească anumite cerinţe, şi anume:

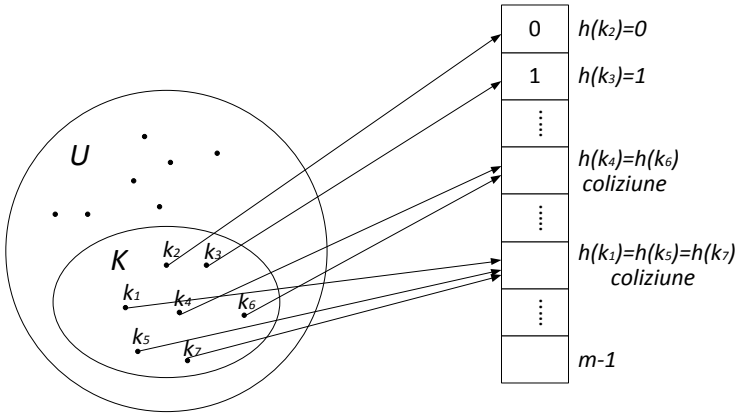
1. Să se poată calcula rapid. Timpul de calcul al lui $h(k)$ ne va da timpul de acces la componenta $T[h(k)]$.
2. Codomeniul ei să fie cât mai mic. Este dezirabil ca m să fie cât mai apropiat de $|K|$, în orice caz diferenţa dintre ele să fie acceptabilă ca dimensiune.
3. h să fie cât mai aproape de o funcţie surjectivă, adică să avem cât mai puţine locaţii goale. Această cerinţă se leagă de precedenta.
4. h să fie cât mai aproape de o funcţie injectivă pe K. Cu alte cuvinte, pentru chei $k_1, k_2 \in K,\ k_1 \neq k_2$ să avem $h(k_1) \neq h(k_2)$.

Numim *coliziune* a cheilor k_1 şi k_2 situaţia în care $h(k_1) = h(k_2)$. Cerinţa 4 se exprimă şi sub forma ,,ne dorim să avem cât mai puţine coliziuni". Din păcate coliziunile nu pot fi evitate complet şi trebuie găsite metode de rezolvare a lor.

Pentru a folosi un tabel de dispersie programatorul trebuie să rezolve două probleme: să aleagă o funcţie de dispersie şi să opteze pentru o metodă de rezolvare a coliziunilor. Aceasta din urmă se împarte în două mari clase :

- rezolvarea coliziunilor prin înlănţuire(când locaţia $T[h(k)]$ conţine o listă înlănţuită a tuturor înregistrărilor cu chei ce au colizionat cu k) şi
- rezolvarea coliziunilor prin adresare directă, când se folosesc metode mai sofisticate de căutare a unei locaţii libere în T pentru o cheie care colizionează.

Ne vom ocupa pe rând de cele 2 tipuri de probleme.



Funcţii de dispersie

Ne ocupăm în acest paragraf de prezentarea câtorva tehnici euristice de creare a unor funcţii de dispersie bune. Am exprimat informal în paragraful precedent câteva cerinţe pentru o asemenea funcţie

$$h : U \to \{0, 1, ..., m-1\}.$$

Vom lucra la **ipoteza hashingului simplu uniform (HSU)**, care, informal, spune că orice cheie este distribuită prin funcţia h cu probabilitate egală în oricare din locaţiile $\{0, \ldots, m-1\}$. Mai precis, pentru oricare $h \in K$ probabilitatea ca $h(k) = i$ este $\frac{1}{m}$ pentru orice $i \in [0 \ldots m-1]$, şi este independentă de restul cheilor. Dacă P este probabilitatea de distrubuţie a cheilor din K în U, adică dacă $P(k)$=probabilitatea de a extrage cheia k din U, atunci ipoteza HSU se formulează:

$$\sum_{\{k|h(k)=i\}} P(k) = \frac{1}{m},\ \forall i = 0, 1, \ldots, m-1$$

Deoarece P este de obicei necunoscută, nu este în general posibil să verificăm că ipoteza HSU este satisfăcută .

În cele ce urmează vom interpreta cheile şi numerele naturale, adică vom considera că $U < N$. Atunci când cheile sunt stringuri, metoda folosită pentru a le converti la numere naturale este urmatoarea: se înlocuieşte fiecare caracter cu codul său ASCII şi se calculează valoarea şirului rezultat ca întreg în baza 128.

De exemplu: AB se scrie $6566_{128} = 65 \cdot 128 + 66 = 8386$ ABC se scrie $656667_{128} = 65 \cdot 128^2 + 66 \cdot 128 + 67 = 1073605$

Metoda diviziuni

Se numesc funcţii de dispersie obţinute prin metoda diviziunii funcţiile de forma

$$h : U \to \{0, \ldots, m-1\},\ h(k) = k \bmod m.$$

Ele sunt printre cele mai comune pentru că sunt uşor de calculat.

Alegerea unei astfel de funcţii revine practic la alegerea lui m, dimensiunea tabelului de dispersie. Deci, ea va fi guvernată, pe de o parte, de dimensiunile lui K şi de modul în care rezolvăm coliziunile. De exemplu, dacă ne propunem să le rezolvăm prin inlănţuire şi să menţinem în medie 3 date într-o locaţie, m ar putea fi apropiat de $\frac{|K|}{3}$. Dacă însă vrem ca toate elementele din K să-şi găsească locul în tabel, atunci am putea alege un m apropiat de $\frac{3|K|}{2}$, acceptând deci o risipă de spaţiu egală cu $\frac{|K|}{2}$.

Dar acestea nu sunt singurele considerente care guvernează alegerea lui m. Important este ca cheile să fie bine dispersate pe mulţimea $\{0, 1, \ldots, m-1\}$ şi să avem cât mai puţine coliziuni. Vom evita valori pentru m de forma lui 2, deoarece, dacă $m = 2$, atunci $h(k) = k \bmod 2^p$ reprezintă doar ultimii p biţi

din scrierea binară a lui k. Am pierdut în felul acesta informaţia conţinută în ceilalţi biţi, lucru nerecomandabil: dacă nu avem indicaţii contrare, trebuie să facem ca $h(k)$ să depindă de toţi biţii lui k. Numerele apropiate de puteri ale lui 2 sunt şi ele de evitat. Un exemplu de rezultat care ne conduce la această decizie este următorul: dacă $m = 2^p - 1$ şi cheile k sunt şiruri de caractere în baza 2^p, atunci 2 şiruri care diferă doar printr-o transpoziţie a două caractere adiacente vor avea aceeaşi valoare prin h.

Vom evita şi valori pentru m de forma puterilor lui 10 , căci dacă am alege un asemenea m, atunci $h(k)$ nu ar ? decât de o parte a caracterelor ce apar în scrierea lui k în baza 10. Valori bune pentru această metodă sunt m, un număr prim şi cât mai departe de puteri ale lui 2. Evident, ne putem propune să indexăm tabelul după mulţimea de indici $\{1, 2, \ldots, m\}$, în caz în care funcţiile din această clasă vor fi de forma $h(k) = k \mathrm{mod} m + 1$.

Să considerăm exemplul firmei cu 68 de angajaţi, identificaţi cu chei întregi cu 4 cifre în scrierea în baza 10. Să presupunem că alegem pentru tabel o dimensiune apropiată de $\frac{3|K|}{2} = \frac{3}{2} \cdot 68$. O bună alegere pentru m ar fi $m = 97$, deoarece este prim , apropiat de $\frac{3}{2} \cdot 68$ şi destul de departe de puteri ale lui 2.

Să considerăm cheile $k_1 = 3205,\ k_2 = 7148$ şi $k_3 = 2345$.

Cu funcţia de dispersie dată de metoda diviziunii

$$h(k) = k \mathrm{mod} 97$$

Obţinem pentru aceste chei următoarele adrese din tabel : $h(k_1) = 3205 \mathrm{mod} 97 = 4$ $h(k_2) = 7148 \mathrm{mod} 97 = 67$ $h(k_3) = 2345 \mathrm{mod} 97 = 17$

Metoda multiplicării

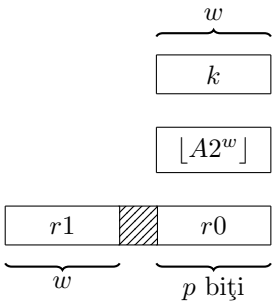
Fie A o constantă pozitivă subunitară, $0 < A < 1$. Să considerăm funcţia dată de

$$h(k) = \lfloor m(kA \mathrm{mod} 1) \rfloor$$

$kA \mathrm{mod} 1$ reprezintă partea fracţionară a lui kA, $kA - [kA]$, care se inmulţeşte cu m, iar rezultatul se trunchiază la cel mai apropiat întreg. Observăm că valoarea lui m nu e critică pentru această metodă, iar puterile lui 2 sunt alegeri bune pentru m, pentru că funcţia se va calcula uşor. Fie $m = 2^p$. Atunci

$$h(k) = \lfloor 2^p(kA - [kA]) \rfloor = \lfloor kA2^p - 2^p[kA] \rfloor$$

Fie w lungimea unui cuvânt maşina în biţi şi presupunem că k se poate reprezenta pe w biţi de forma



$K\ \lfloor A2^w \rfloor = r_1 2^w + r_0$

Deoarece $kA2^p = (kA2^w)\, 2^{p-w}$ înseamnă că $h(k) = (r_1 2^w + r_0)2^{p-w}$.

Knuth recomandă pentru A numărul de aur, $\frac{\sqrt{5}+1}{2}$, a cărui valoare aproximată cu 7 zecimale este $1,6180339$. Revenind la exemplul nostru, să alegem pentru această metodă $A = 0,6180339$[1] şi $m = 2^6 = 64$. Atunci, pentru cele 3 chei din exemplu şi funcţia $h(k) = \lfloor m(kAmod1) \rfloor$, avem: $h(k_1) = h(3205) = 51$ deoarece partea fracţională a lui $3205 \cdot A$ este $0,7986$, care, înmulţit cu 64 ne dă $51,1107$, a cărui parte întreagă este 51. Analog, obţinem $h(k_2) = h(7148) = 45$ şi $h(k_3) = h(2345) = 18$.

Metoda împachetării

Metoda împachetării sau folding generează funcţii de dispersie în felul următor. Fiecare cheie k, despre care am făcut presupunerea că este întreg, este partiţionată în bucăţi de lungimi egale (eventual cu excepţia ultimei), $k_1,\ k_2,\ \ldots,\ k_r$. Dacă k este un întreg scris în baza 10 bucăţile vor fi şiruri de caractere în baza 10, de aceeaşi lungime fixă, l, pe care din nou le putem considera întregi. O funcţie de dispersie

$$h : U \to [0 \ldots 10^2 - 1]$$

este dată de formula:

$$h(k) = (k_1 + k_2 + \cdots + k_r) \mathrm{mod} 10^l$$

ceea ce înseamnă că se adună cele r bucăţi, ca întregi, şi din sumă se păstrează doar l cifre, cele mai puţin semnificative. Alte funcţii de dispersie se obţin inversând ordinea cifrelor în toate bucăţile pare sau în toate bucăţile impare. De exemplu:

$$h(k) = (\overline{k_1} + k_2 + \overline{k_3} + \ldots) \mathrm{mod} 10^l$$

unde k este întregul obţinut prin inversarea ordinii caracterelor. Pe exemplul nostru metoda împachetării se aplică în felul următor: spargem fiecare cheie de

[1] Pentru calcularea valorii $h(k)$ nu păstrăm decât partea fracţionară a produsului kA. Prin urmare, este suficient să alegem A ca fiind partea fracţionară a numărului $\frac{\sqrt{5}+1}{2}$, adică $0,6180339$, care este chiar $\frac{\sqrt{5}+1}{2}$.

4 caractere în 2 bucăţi de câte 2 caractere, pe care le adunăm şi omitem, dacă e cazul, cifra cea mai nesemnificativă pentru ca rezultatul să aibă lungimea tot de 2 caractere. Avem deci $h(k_1) = h(3205) = 32 + 5 = 37$ $h(k_2) = h(7148) = (71 + 48)\text{mod}100 = 19$ $h(k_3) = h(2345) = 23 + 45 = 68$ Pentru funcţia $h'(k) = k_1 + \overline{k_2}$, în care a doua funcţie o inversăm în oglindă înainte de adunare, obţinem $h'(3205) = 32 + \overline{05} = 32 + 50 = 82$ $h'(7148) = 71 + \overline{48} = (71 + 84)\text{mod}100 = 55$ $h'(2345) = 23 + \overline{45} = 23 + 54 = 77$

Metoda pătratului

Presupunem că l este lungimea fixată a lui $h(k)$ ca întreg în baza 10. Fie

$$h : U \to [0 \dots 10^l],\ h(k) = (k^2 \text{div} 10^{c_1}) \text{mod} 10^{c_2}.$$

Se ridică la pătrat, iar din întregul lung astfel obţinut se elimină c_1 cifre dintre cele mai puţin semnificative si c_2 cifre dintre cele mai semnificative, păstrându-se exact l cifre din ,,centrul" lui k^2. Cu alte cuvinte c_1 şi l sunt fixate, iar $c_@$ se obţine din formula:

$$l = lung(k^2) - c_1 - c_2.$$

Să aplicăm această metodă exemplului nostru. O cheie de 4 cifre va avea pătratul de 7 sau 8 cifre, dintre care să eliminăm $c_1 = 3$ dintre cele din dreapta, să păstrăm $l = 2$ cifre, şi să eliminăm şi restul de cifre ,,centrale", adică a 4-a şi a 5-a din stânga. $k_1 = 3205$: ${k_1}^2 = 3205^2 = 102\underline{72}025$ şi $h(k_1) = 72$ $k_2 = 7148$: ${k_2}^2 = 7148^2 = 510\underline{93}904$ şi $h(k_2) = 93$ $k_3 = 2345$: ${k_3}^2 = 2345^2 = 54\underline{99}025$ şi $h(k_3) = 99$