Hash-uri

Tabele cu adresare directă. Tabele de dispersie

Prewatch

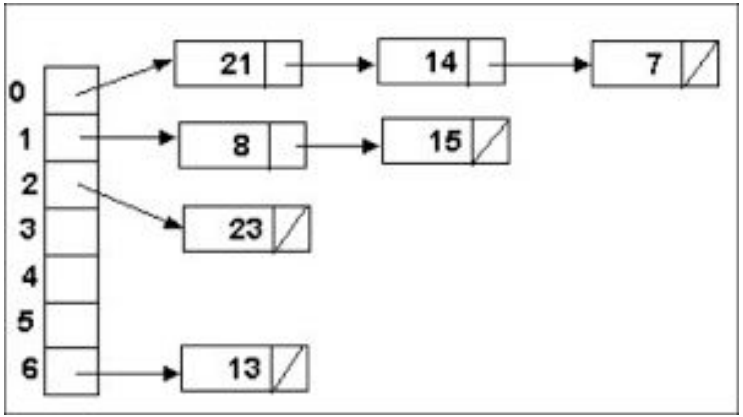
Prewatch:

- Video1 (pentru sync-video: https://youtu.be/JZHBa-rLrBA)
  - Ideal tot videoul
  - **Măcar de la 29 la 1:02 (33 de minute)**
- Video2 (pentru sync-video: https://youtu.be/0M_kIqhwbFo)
  - Ideal tot videoul :)

Funcții de dispersie

Funcții de dispersie:

- Am zis săptămâna trecută că, pentru moment, folosim `h(x) = x % p`, unde p este un număr prim.



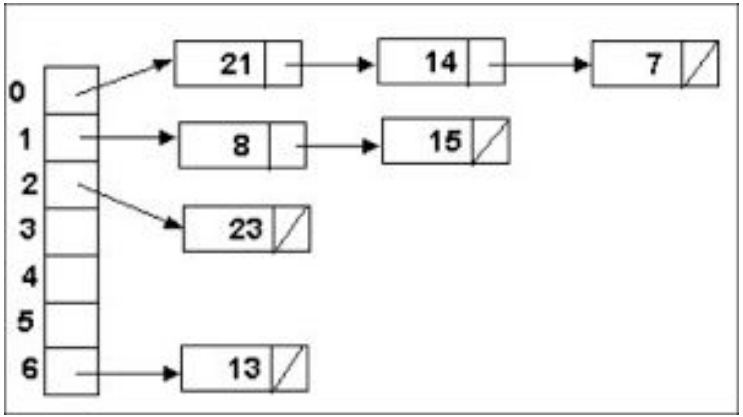
Rezolvarea coliziunilor:

- Am spus că vom ține o listă înlănțuită

Funcții de dispersie

Funcții de dispersie:

- Am zis săptămâna trecută că, pentru moment, folosim `h(x) = x % p`, unde p este un număr prim.



Rezolvarea coliziunilor:

- Am spus că vom ține o listă înlănțuită

- Complexitate:
  - O(1) căutare ?
  - Ce se întâmplă dacă p este ~ sqrt(n)?
    - O(sqrt n) pe căutare
    - Dacă datele nu sunt rele avem O(n/p) ... p nu trebuie să fie mult mai mic decât n, ideal mai mare

Funcții de dispersie

Funcții de dispersie:

- Am zis săptămâna trecută că, pentru moment, folosim `h(x) = x % p`, unde p este un număr prim.

- Ce ne dorim de la o funcție hash? **Ipoteza dispersiei uniforme simple:**
  - Fiecare cheie se poate dispersa cu aceeași probabilitate în oricare din cele m locații.
  - f(x) = cel mai reprezentativ bit a lui x - nu e bună
    - f(24) = f(18) = 16 → cheile nu au aceeași probabilitate să ajungă pe cele m locații
  - În practică, nu putem satisface perfect regula, dar ne dorim să fim cât mai aproape
- https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-046j-introduction-to-algorithms-sma-5503-fall-2005/video-lectures/lecture-7-hashing-hash-functions/ (28:38)
- https://drive.google.com/drive/u/0/folders/1aNqJk0kfKZszEOzvPLh81hvH7OSKbI1g

Funcții de dispersie

Funcții de dispersie:

- Am zis săptămâna trecută că, pentru moment, folosim `h(x) = x % p`, unde p este un număr prim.

- https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-046j-introduction-to-algorithms-sma-5503-fall-2005/video-lectures/lecture-7-hashing-hash-functions/ (28:38) (recomandare)

- Am vorbit despre Funcții de dispersie și ne-am uitat la diverse metode
  - Metoda diviziunii (discutată și data trecută)
  - Metoda multiplicării (folosită în practică mult, pentru că este mai rapidă)

Dispersie universală

Fie **H** o colecţie finită de funcţii de dispersie, care transformă un univers dat **U** al cheilor, în domeniul **{0, 1, ..., m−1}**.

O astfel de colecţie se numeşte **universală** dacă, pentru fiecare pereche de chei distincte **x, y ∈ U**, numărul de funcţii de dispersie **h ∈ H** pentru care h(x) = h(y) este exact **|H| / m**.

Cu alte cuvinte, cu o funcţie de dispersie aleasă aleator din H, şansa unei coliziuni între x şi y când **x != y** este exact 1/m, care este exact şansa unei coliziuni dacă h(x) şi h(y) sunt alese aleator din mulţimea {0, 1, ..., m − 1}.

Dispersie universală

Următoarea teoremă arată că o clasă universală de funcţii de dispersie dă un comportament bun în cazul mediu.

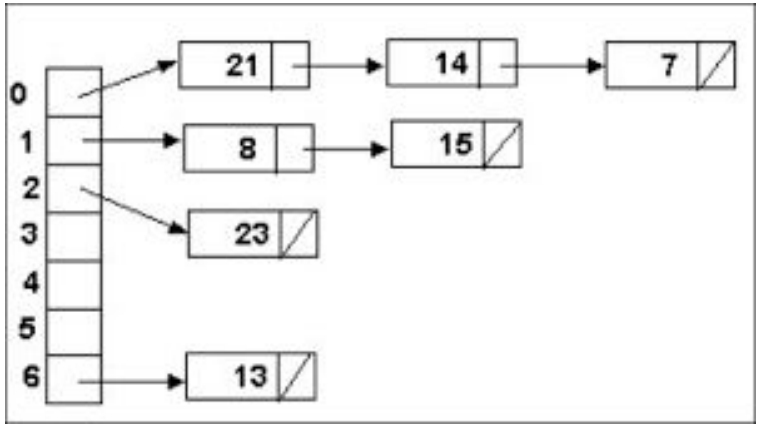
Teorema 6.1: Dacă **h** este aleasă dintr-o colecţie universală de funcţii de dispersie şi este folosită pentru a dispersa **n** chei într-o tabelă de dimensiune **m**, unde **n ≤ m**, numărul mediu de coliziuni în care este implicată o cheie particulară **x** este mai mic decât 1.

Rezolvarea coliziunilor

Rezolvarea coliziunilor

- Am zis săptămâna trecută că, pentru moment, folosim **înlănțuirea**

- https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-046j-introduction-to-algorithms-sma-5503-fall-2005/video-lectures/lecture-7-hashing-hash-functions/ (50:00)



Rezolvarea coliziunilor

Rezolvarea coliziunilor

- Am zis săptămâna trecută că, pentru moment, folosim **înlănțuirea**
- Am urmărit 15 minute din cursul de la MIT, pornind cu 50:00, unde se vorbește despre metoda adresării directe
- În cazul adresării directe, au fost evidențiate 2 metode de calculare a poziției elementului în tabelul de dispersie de mărime m:
  - **Testare liniară:** `h(x,i) = (h(x,0) + i) % m`
  - **Hash dublu:** `h(x,i) = (h1(x) + i * h2(x)) % m`

Rezolvarea coliziunilor

Altă metodă:

- https://en.wikipedia.org/wiki/Cuckoo_hashing