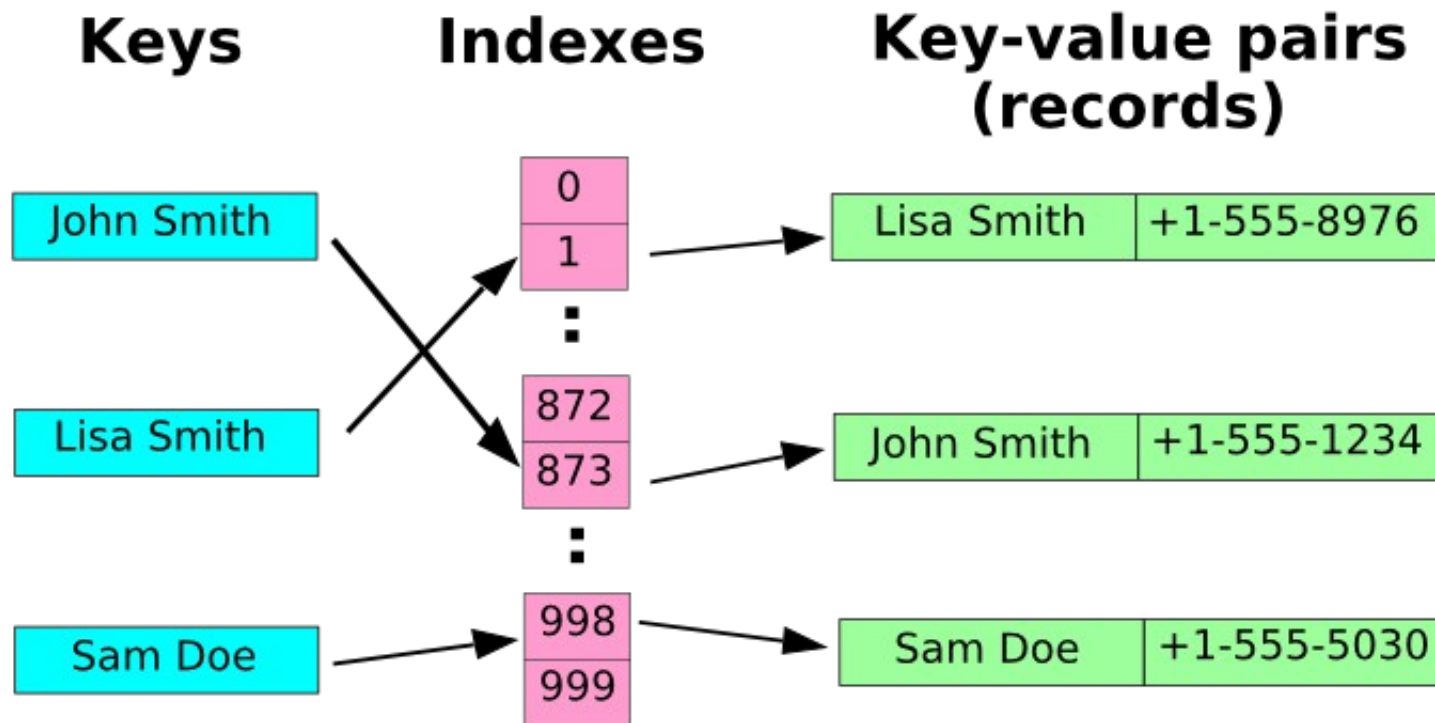


# Transformacje kluczy (*hashing*)

- Zamiast stosować skomplikowane struktury staramy się znaleźć położenie danych na podstawie samego klucza, najlepiej w jednym kroku – złożoność  $O(1)$ .
- Dane przechowujemy w tablicy, umiejscowienie danego elementu znajdujemy wyznaczając wartość funkcji mieszającej (*hash function*):  
 $H: K \rightarrow A$       K-klucze      A-adresy
- Przykład: książka adresowa, lista numerów PESEL dla 10000 osób ( $10^{11} \rightarrow 10^5$ )

# Transformacje kluczy (*hashing*)



*Źródło: wikipedia.org*

# Transformacje kluczy (*hashing*)

- Kolizja – przypisanie przez funkcję mieszającą tej samej wartości dwóm różnym kluczom.
- Doskonała funkcja mieszająca – nigdy nie powoduje kolizji.
- Efekt lawiny (*avalanche effect*) – zmiana jednego bitu w kluczu, powoduje zmianę co najmniej połowy bitów w wyniku funkcji.
- Przykładowa funkcja:  $H(k) = \text{ORD}(k) \text{ MOD } n$   
ORD(k) – numer porządkowy klucza  
n – wielkość tablicy

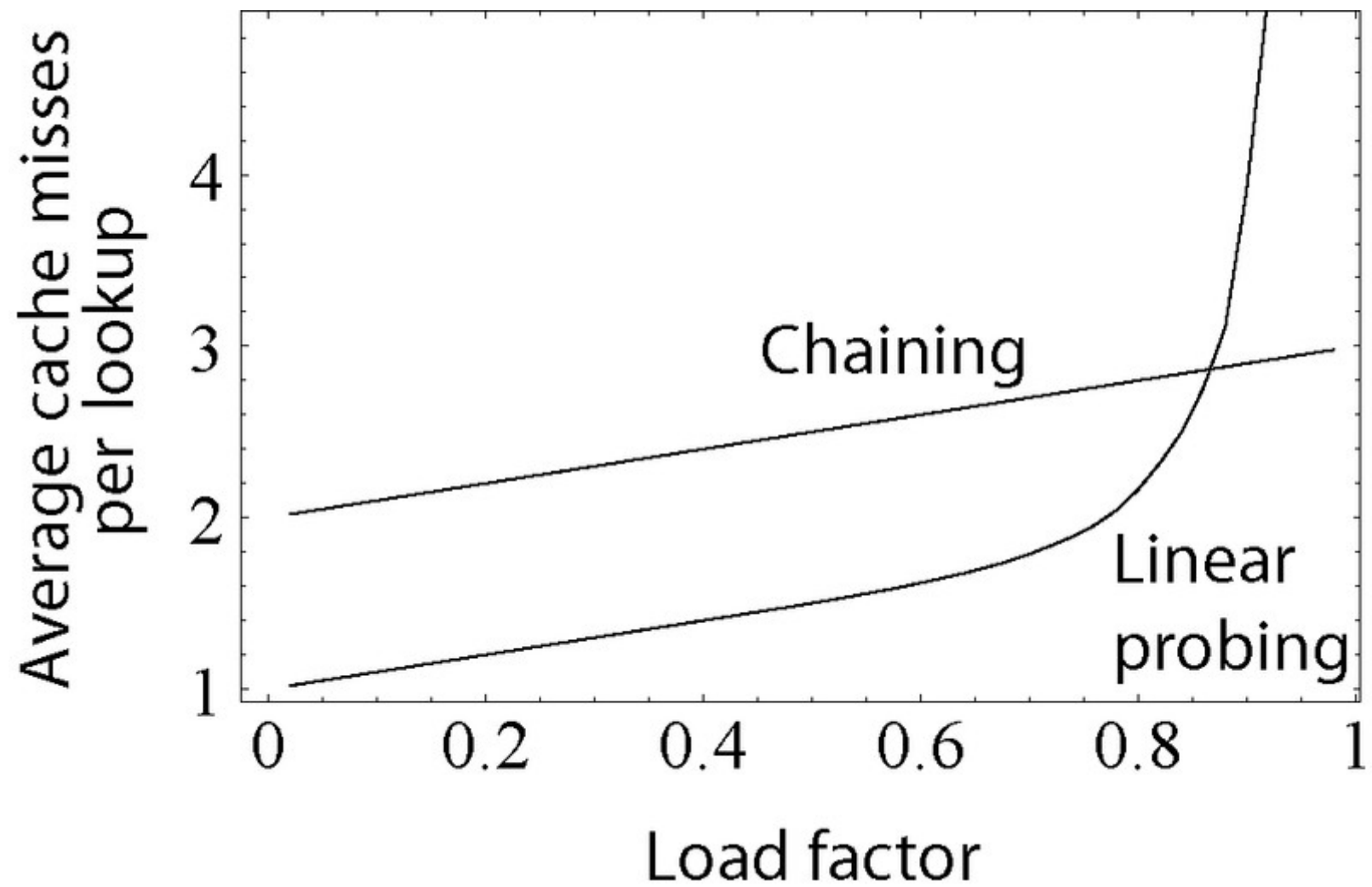
# Rozwiązywanie kolizji

- Metoda łańcuchowa – tworzymy listę dla każdego indeksu.
- Adresowanie otwarte – podczas dodawania, jeśli pod indeksem  $h_0 = H(k)$  są już dane, wyliczamy kolejny adres  $h_1$ , ewentualnie kolejny  $h_2$ , itd. Nowa lokalizacja określana jest przez dodanie do wartości funkcji mieszającej wartości tzw. funkcji przyrostu  $p(i)$ , gdzie  $i$  oznacza numer próby, a więc:

$$h_{i+1} = [h_0 + p(i)] \bmod n$$

# Adresowanie otwarte

- szukanie liniowe - funkcja przyrostu postaci  $p(i)=i$
- szukanie kwadratowe -  $p(i)=i^2$ ; aby zmniejszyć złożoność możemy skorzystać z zależności:  
$$h_{i+1}=h_i+d_i \quad d_{i+1}=d_i+2 \quad h_0=0 \quad d_0=1$$
- mieszanie podwójne -  $p(i)=i*h'(k)$ , gdzie  $h'$  jest dodatkową funkcją mieszającą od klucza  $k$ .



*Źródło: wikipedia.org*

# Analiza (za N.Wirth)

- $a = m/(n+1)$
- $a$  - współczynnik zajętości (*load factor*)  
m-ilość kluczy w tabeli
- $E$  – średnia oczekiwana ilość prób potrzebnych do wstawienia klucza
- Dla funkcji rozrzucającej klucze idealnie losowo:  
 $E_I = -\ln(1-a)/a$
- Dla szukania liniowego:  
 $E_L = (1-a/2) / (1-a)$

# Analiza (za N.Wirth)

<b>a</b>	<b>E<sub>I</sub></b>	<b>E<sub>L</sub></b>
0.10	1.05	1.06
0.25	1.15	1.17
0.50	1.39	1.50
0.75	1.85	2.50
0.90	2.56	5.50
0.95	3.15	10.50



- Złożoność pesymistyczna jest liniowa  $O(n)$ , ale średnia jest  $O(1)$  - niezależna od wielkości tablicy.
- Obliczenie dobrej funkcji mieszającej może być kosztowne.
- Pamięć podręczna przyspiesza odwołania do komórek pamięci operacyjnej gdy są one zgrupowane blisko siebie, zastosowanie tablicy mieszającej dla zbyt małej liczby elementów może być wolniejsze niż zastosowanie zwykłej tablicy przeszukiwanej sekwencyjnie.
- Usuwanie jest problematyczne.

# Usuwanie z tablicy haszującej

- Zamiast usuwać dany element, zaznaczamy że komórka w tablicy jest już wolna. Dzięki temu, nie skończymy na niej wyszukiwania, przy następnej operacji dodawania lub wyszukiwania.