

Sia \mathcal{U} un insieme detto *universo* con $|\mathcal{U}| = m \gg 1$. Consideriamo una tabella hash T di dimensione $1 < n \ll m$ ed una funzione di hash $h : \mathcal{U} \rightarrow \{0, \dots, n-1\}$. Supponiamo di memorizzare un arbitrario sottoinsieme $S \subseteq \mathcal{U}$ nella tabella hash usando la funzione h . Ovvero, memorizziamo ogni $u \in S$ nella posizione $h(u)$ di T . Dato che la tabella hash è molto più piccola della cardinalità di \mathcal{U} , potranno avvenire delle collisioni. Una *collisione* consiste nell'evento in cui due elementi distinti $u, v \in S$ sono tali che $h(u) = h(v)$. Ciò significa che u e v andrebbero memorizzati nella stessa posizione della tabella hash T . Questo problema viene solitamente gestito utilizzando, per esempio, delle liste associate ad ogni posizione di T . Una conseguenza di questa soluzione alle collisioni riguarda il costo delle operazioni sulla tabella hash: ogni ulteriore operazione di ricerca o inserimento sulla tabella richiederà tempo proporzionale alla lunghezza della lista associata alla posizione della tabella dove viene fatta l'operazione.

In un'analisi di caso peggiore, il tempo per eseguire un'operazione di ricerca o inserimento di un elemento $v \in \mathcal{U}$ è quindi proporzionale al massimo numero ℓ_{\max} di elementi in S diversi da v che sono stati mappati da h nella stessa posizione di v in T . Più precisamente,

$$\ell_{\max} = \max_{v \in \mathcal{U}} |\{u \in S : h(u) = h(v), u \neq v\}|.$$

Idealmente, vorremmo avere $\ell_{\max} = \mathcal{O}(|S|/n)$, in modo da minimizzare il tempo di esecuzione di un'operazione nel caso peggiore.

Possiamo ottenere questo risultato usando la randomizzazione. Supponiamo di avere a disposizione una famiglia \mathcal{H} di funzioni $h : \mathcal{U} \rightarrow \{0, \dots, n-1\}$, e di usare una funzione di hash h che sia la realizzazione di una variabile casuale H corrispondente all'estrazione casuale (con probabilità uniforme) di un elemento da \mathcal{H} . Supponiamo ora che \mathcal{H} soddisfi la condizione seguente

$$\mathbb{P}(H(u) = H(v)) = \frac{1}{n} \quad \text{per ogni } u, v \in \mathcal{U}, u \neq v \quad (1)$$

dove la probabilità è calcolata rispetto all'estrazione casuale della funzione di hash H da \mathcal{H} .

La condizione (1) è sufficiente a garantire che

$$\max_{v \in \mathcal{U}} \mathbb{E} \left[|\{u \in S : H(u) = H(v), u \neq v\}| \right] \leq \frac{|S|}{n}$$

dove il valore atteso è calcolato rispetto all'estrazione di H da \mathcal{H} . Infatti, supponiamo di aver estratto H a caso da \mathcal{H} e di averla usata per inserire $S = \{u_1, \dots, u_s\}$ nella tabella. Sia $v \in \mathcal{U}$ un elemento arbitrario che vogliamo cercare o inserire nella tabella. Definiamo le variabili casuali X_1, \dots, X_s dove $X_i = \mathbb{I}\{H(u_i) = H(v)\}$ e $\mathbb{I}\{\cdot\}$ è la funzione indicatrice di un evento, definita come

$$\mathbb{I}\{A\} = \begin{cases} 1 & \text{se } A \text{ è vero} \\ 0 & \text{altrimenti.} \end{cases}$$

Si ricorda che una proprietà della funzione indicatrice di un evento A è che il suo valore atteso equivale a $\mathbb{E}[\mathbb{I}\{A\}] = \mathbb{P}(A)$. Allora,

$$|\{u \in S : H(u) = H(v), u \neq v\}| = \sum_{\substack{i=1 \\ u_i \neq v}}^s X_i .$$

Il valore atteso di questo numero è calcolato come

$$\begin{aligned} \mathbb{E} \left[\sum_{\substack{i=1 \\ u_i \neq v}}^s X_i \right] &= \sum_{\substack{i=1 \\ u_i \neq v}}^s \mathbb{E}[X_i] && \text{(per linearità del valore atteso)} \\ &= \sum_{\substack{i=1 \\ u_i \neq v}}^s \mathbb{P}(H(u_i) = H(v)) && \text{(per definizione di } X_i) \\ &= \sum_{\substack{i=1 \\ u_i \neq v}}^s \frac{1}{n} && \text{(per ipotesi su } \mathcal{H}, \text{ eq. (1))} \\ &\leq \frac{|S|}{n} . && \text{(al più } |S| \text{ elementi nella sommatoria)} \end{aligned}$$

Non è difficile trovare famiglie \mathcal{H} che soddisfano la condizione (1). Assumiamo per semplicità che $\mathcal{U} = \{0, \dots, m-1\}$ e consideriamo la classe \mathcal{H} di tutte le funzioni $h : \{0, \dots, m-1\} \rightarrow \{0, \dots, n-1\}$. Ognuna di queste funzioni di hash corrisponde a un vettore $\mathbf{h} = (h_0, \dots, h_{m-1}) \in \{0, \dots, n-1\}^m$ in modo che $h(u) = h_u$. Allora, il numero di vettori $\mathbf{h} \in \{0, \dots, n-1\}^m$ tali che $h_u = h_v$ è n^{m-1} , per $u, v \in \mathcal{U}$ distinti. Quindi, se estraggo H uniformemente a caso da \mathcal{H} ,

$$\mathbb{P}(H(u) = H(v)) = \frac{|\{\mathbf{h} \in \mathcal{H} : h_u = h_v\}|}{|\mathcal{H}|} = \frac{n^{m-1}}{n^m} = \frac{1}{n}$$

e la condizione (1) è soddisfatta. D'altra parte, la classe \mathcal{H} non è utilizzabile in pratica in quanto mi servono $\lceil \log_2 |\mathcal{H}| \rceil = \Theta(m \log n)$ bit per memorizzare ciascuna $h \in \mathcal{H}$ e stiamo assumendo che $m \gg 1$.

Per eliminare situazioni di questo genere, aggiungiamo alla condizione (1) una richiesta ulteriore (che denoteremo successivamente come proprietà 2), ovvero che ogni $h \in \mathcal{H}$ possa essere rappresentata con al più $\Theta(\log m)$ bit (che è anche lo spazio che mi occorre per rappresentare un elemento arbitrario di \mathcal{U}) e calcolata in modo efficiente. Una famiglia \mathcal{H} di funzioni $h : \mathcal{U} \rightarrow \{0, \dots, n-1\}$ che soddisfa entrambe queste condizioni è detta una *famiglia universale di funzioni di hash*.

Dimostriamo ora l'esistenza di famiglie universali. Oltre a $\mathcal{U} = \{0, \dots, m-1\}$ assumiamo anche $n = p$ per un qualche p primo. Rappresentiamo ciascun elemento $u \in \{0, \dots, m-1\}$ come un numero $[u]_p$ in base p . Più precisamente, $[u]_p = x = (x_1, \dots, x_r)$ dove $x_i \in \{0, \dots, p-1\}$ e r è il più piccolo intero tale che $p^r \geq m$. Ovvero,

$$r = \left\lceil \frac{\log_2 m}{\log_2 p} \right\rceil .$$

Sia $\mathcal{A} = \{0, \dots, p-1\}^r$ l'insieme usato per rappresentare gli elementi di \mathcal{U} . Introduciamo ora la famiglia di funzioni di hash $\mathcal{H} = \{h_a : a \in \mathcal{A}\}$ di tipo $h_a : \mathcal{A} \rightarrow \{0, \dots, p-1\}$ e definite come

$$h_a(u) = \left(\sum_{i=1}^r a_i x_i \right) \pmod{p} \quad \text{dove } x = [u]_p.$$

Si noti che posso rappresentare ogni h_a con

$$|\log_2 |\mathcal{A}|| = \Theta(r \log p) = \Theta\left(\frac{\log m}{\log p} \log p\right) = \Theta(\log m)$$

bit e posso calcolare h_a in modo efficiente. Per verificare la condizione (1) faremo uso del lemma seguente.

Lemma 1 *Per ogni primo p , per ogni α, β e z interi,*

$$z \not\equiv 0 \pmod{p} \quad \text{e} \quad \alpha z \equiv \beta z \pmod{p} \quad \text{implicano} \quad \alpha \equiv \beta \pmod{p}.$$

DIMOSTRAZIONE. Chiaramente, $\alpha z \equiv \beta z \pmod{p}$ se e solo se $z(\alpha - \beta) \equiv 0 \pmod{p}$. Inoltre, $z \not\equiv 0 \pmod{p}$ e $z(\alpha - \beta) \equiv 0 \pmod{p}$ implicano $(\alpha - \beta) \equiv 0 \pmod{p}$. Infatti, se p è primo e z non contiene p come fattore, allora $\alpha - \beta$ lo deve contenere (se invece p non fosse primo, allora z e $\alpha - \beta$ potrebbero spartirsi i fattori primi di p). \square

Siamo ora pronti a dimostrare che per \mathcal{H} vale la proprietà 1. Dato che, come abbiamo già osservato, per la stessa classe valeva anche la proprietà 2, concludiamo che \mathcal{H} è una famiglia universale di funzioni di hash.

Teorema 2 *\mathcal{H} è tale che per ogni $u, v \in \mathcal{U}$ distinti, la frazione di elementi $a \in \mathcal{A}$ tali che $h_a(u) = h_a(v)$ è al più $\frac{1}{p}$. Quindi, se H è estratta a caso da \mathcal{H} , allora*

$$\mathbb{P}(H(u) = H(v)) \leq \frac{1}{p}.$$

DIMOSTRAZIONE. Siano $x = [u]_p$ e $y = [v]_p$. Sappiamo che $x \neq y$ poiché $u \neq v$. Allora, esiste almeno una coordinata $j \in \{1, \dots, r\}$ tale che $x_j \neq y_j$. Per estrarre H a caso in \mathcal{H} prendiamo $a \in \mathcal{A}$ estraendo a caso ciascuna coordinata $a_i \in \{0, \dots, p-1\}$. Per qualunque estrazione dei valori a_i sulle coordinate $i \neq j$, abbiamo che

$$\begin{aligned} h_a(u) = h_a(v) &\iff \left(\sum_{i=1}^r a_i x_i \equiv \sum_{i=1}^r a_i y_i \right) \pmod{p} \\ &\iff a_j \underbrace{(y_j - x_j)}_z \equiv \underbrace{\sum_{i:i \neq j} a_i (x_i - y_i)}_k \pmod{p} \\ &\iff a_j z \equiv k \pmod{p}. \end{aligned}$$

Dimostriamo ora che questa congruenza può essere soddisfatta da non più di un valore di a_j . In altre parole, per qualunque scelta dei valori di $a_i \in \{0, \dots, p-1\}$ con $i \neq j$, esiste al più un unico valore di $a_j \in \{0, \dots, p-1\}$ tale che $a_j z \equiv k \pmod{p}$ sia vera. Per assurdo, supponiamo che esistano a_j, a'_j valori distinti per la coordinata j che soddisfino entrambi la congruenza. Allora avremmo anche che $a_j z \equiv a'_j z \pmod{p}$. Dato che $z \not\equiv 0 \pmod{p}$ per ipotesi (perché $x_j \neq y_j$ e $0 \leq x_j, y_j < p$), il Lemma 1 implica che $a_j \equiv a'_j \pmod{p}$. Siccome $0 \leq a_j, a'_j < p$, ciò significa che $a_j = a'_j$ e si ha una contraddizione. Quindi, c'è al più un solo valore della coordinata j che può rendere $h_a(u) = h_a(v)$ vera e la probabilità di estrarre questo valore fra i p possibili per a_j è al più $\frac{1}{p}$. \square

Nota a margine: per maggiore chiarezza, potremmo argomentare più formalmente per arrivare alla medesima conclusione della dimostrazione del Teorema 2. Seguendo lo stesso ragionamento, sappiamo che

$$\{h_a : h_a(u) = h_a(v), a \in \mathcal{A}\} = \{h_a : a_j z \equiv k \pmod{p}, a \in \mathcal{A}\}$$

per via dell'equivalenza tra le due condizioni, e che dunque

$$\begin{aligned} |\{h_a : h_a(u) = h_a(v), a \in \mathcal{A}\}| &= |\{h_a : a_j z \equiv k \pmod{p}, a \in \mathcal{A}\}| \\ &= \sum_{a \in \mathcal{A}} \mathbb{I}\{a_j z \equiv k \pmod{p}\} \\ &= \sum_{\substack{a_i \in \{0, \dots, p-1\} \\ i \neq j}} \underbrace{\sum_{a_j \in \{0, \dots, p-1\}} \mathbb{I}\{a_j z \equiv k \pmod{p}\}}_{\leq 1} \\ &\leq \sum_{\substack{a_i \in \{0, \dots, p-1\} \\ i \neq j}} 1 \\ &= |\{(a_1, \dots, a_{j-1}, a_{j+1}, \dots, a_r) : a_i \in \{0, \dots, p-1\}, \forall i \neq j\}| \\ &= p^{r-1}, \end{aligned}$$

dove la disuguaglianza è dovuta all'esistenza di al più un valore di a_j che soddisfa la congruenza. Quindi, ricordando che $|\mathcal{H}| = |\mathcal{A}| = p^r$, la frazione di elementi $a \in \mathcal{A}$ tali che $h_a(u) = h_a(v)$ è

$$\frac{|\{h_a : h_a(u) = h_a(v), a \in \mathcal{A}\}|}{|\mathcal{H}|} \leq \frac{p^{r-1}}{p^r} = \frac{1}{p}.$$