Complementi di Algoritmi e Strutture Dati

Hashing Universale

Docenti: *Nicolò Cesa-Bianchi* versione 29 marzo 2026

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 $h : \mathcal{U} \to \{0, \dots, n-1\}$. 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 di un elemento $v \in \mathcal{U}$ è quindi proporzionale al massimo numero $\ell_{\max}$ di elementi in S che sono stati mappati da h nella stessa posizione di v in T. Più precisamente,

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

Idealmente, vorremmo avere $\ell_{\max} = \mathcal{O}\big(|S|/n\big)$, 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 $h : \mathcal{U} \to \{0, \dots, n-1\}$, 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

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

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

$$\max_{S \subset \mathcal{U} \,:\, |S| = N} \max_{v \in \mathcal{U}} \mathbb{E}\Big[\big|\{u \in S \,:\, H(u) = H(v)\}\big|\Big] = \frac{N}{n}$$

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,

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

Il valore atteso di questo numero è calcolato come

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

Non è difficile trovare famiglie $\mathcal{H}$ che soddisfano la condizione (1). Assumiamo per semplicità che $\mathcal{U} = \{0, \ldots, m-1\}$ e consideriamo la classe $\mathcal{H}$ di tutte le funzioni $h : \{0, \ldots, m-1\} \to \{0, \ldots, n-1\}$. Ognuna di queste funzioni di hash corrisponde a un vettore $\boldsymbol{h} = (h_0, \ldots, h_{m-1}) \in \{0, \ldots, n-1\}^m$ in modo che $h(u) = h_u$. Allora, il numero di vettori $\boldsymbol{h} \in \{0, \ldots, n-1\}^m$ 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 $\mathcal{H}$,

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

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 $h : \mathcal{U} \to \{0, \ldots, n-1\}$ che soddisfa entrambe queste condizioni è detta una *famiglia universale di funzioni di hash*.

Dimostriamo ora l'esistenza di famiglie universali. Oltre a $\mathcal{U} = \{0, \ldots, m-1\}$ assumiamo anche $n = p$ per un qualche p primo. Rappresentiamo ciascun elemento $u \in \{0, \ldots, m-1\}$ come un numero $[u]_p$ in base p. Più precisamente, $[u]_p = x = (x_1, \ldots, x_r)$ dove $x_i \in \{0, \ldots, p-1\}$ 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 $\mathcal{A} = \{0, \ldots, p-1\}^r$ 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 $h_a : \mathcal{A} \to \{0, \ldots, p-1\}$ e definite come

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

Si noti che posso rappresentare ogni h_a con

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

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

Lemma 1 *Sia*

$$c_1 x_1 + \cdots + c_k x_k \equiv b \mod p$$

un'equazione lineare, dove p è primo e non tutti i coefficienti $c_1, \ldots, c_k$ sono uguali a zero. Allora l'equazione ha esattamente p^{k-1} soluzioni in $\{0, 1, \ldots, p-1\}$.

Dimostrazione. Sia j tale che $c_j \neq 0$ (ce n'è almeno uno). Dato che p è primo, ogni elemento di $\{0, 1, \ldots, p-1\}$ ha un'inversa moltiplicativa. Quindi c_j^{-1} esiste e possiamo risolvere l'equazione per x_j,

$$x_j \equiv c_j^{-1} \left(b - \sum_{i \neq j} c_i x_i \right) \mod p$$

Per ogni possibile scelta delle variabili $\{x_i : i \neq j\}$, la formula sopra ha una soluzione. Dato che le scelte possibili per ciascuna variabile sono p, il numero totale di soluzioni è esattamente p^{k-1}. □

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*

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

Dimostrazione. Siano $x = [u]_p$, $y = [v]_p$ e sia $d = x - y$. Dato che $u \neq v$, $x \neq y$ e quindi non tutte le coordinate di d sono identicamente nulle. Per estrarre H a caso in $\mathcal{H}$ prendiamo $a \in \mathcal{A}$ estraendo a caso ciascuna coordinata $a_i \in \{0, \ldots, p-1\}$. Per qualunque estrazione di a, abbiamo che

$$\begin{aligned} h_a(u) = h_a(v) &\iff \left(\sum_{i=1}^{r} a_i x_i \equiv \sum_{i=1}^{r} a_i y_i \right) \mod p \\ &\iff \left(\sum_{i=1}^{r} a_i (x_i - y_i) \right) \equiv 0 \mod p \\ &\iff \left(\sum_{i=1}^{r} a_i d_i \right) \equiv 0 \mod p \end{aligned}$$

Quindi la probabilità di una collisione è esattamente la frazione di vettori $a \in \mathcal{A}$ che sono soluzioni dell'equazione lineare $a_1 d_1 + \cdots + a_r d_r = 0 \mod p$. Dato che d non è un vettore identicamente nullo, usando il Lemma 1 concludiamo che il numero di tali soluzioni è esattamente p^{r-1}. Quindi

$$\mathbb{P}\big(H(u) = H(v)\big) = \frac{p^{r-1}}{|\mathcal{A}|} = \frac{p^{r-1}}{p^r} = \frac{1}{p}$$

che conclude la dimostrazione. □