Relazione esercizio 3: hashmap

Studente: Cagnazzo Christian Damiano – Matricola: 883100

Il caricamento all'interno dell'hashmap di tutte le coppie chiave-valore impiega circa 5 secondi. Questo perché la capacità iniziale è impostata a 8 e quindi è richiesto un numero elevato di ridimensionamenti della struttura con conseguente riposizionamento. Se l'utente, invece, scegliesse come capacità un valore più grande del numero di coppie da inserire (possibilmente lontano da una potenza di 2 per evitare molte collisioni) evitando quindi un ridimensionamento otterremmo un tempo di circa 3 secondi per il caricamento. Difatti l'inserimento in un'hashmap richiede tempo costante se si suppone l'assenza di ridimensionamenti e doppioni (evitando quindi un controllo) in quanto viene effettuato in testa a liste doppiamente concatenate.

Anche il caricamento in un array statico impiega tempo costante ed è leggermente più veloce in quanto si effettua un accesso diretto e non viene effettuato un controllo di eventuali doppioni. Esso impiega nel nostro caso circa un secondo più un altro secondo per ordinare gli elementi rispetto le chiavi, ordinamento effettuato dopo l'inserimento tramite il quick sort che richiede tempo O(nlogn).

Il vantaggio dell'utilizzo di una hasmap si ritrova però nel recuperare i valori corrispondenti alle chiavi. Il recupero di un valore all'interno di un array ordinato tramite una ricerca dicotomica richiede tempo O(logn): all'interno della nostra app tutte le chiavi vengono recuperate in circa 6 secondi. Al contrario, il recupero all'interno di un'hashmap con poche collisioni richiede in media tempo O(1), in quanto tramite la funzione hash si accede quasi in diretto (o quasi in caso di collisione) al nodo interessato. Infatti, tutte le chiavi all'interno della nostra applicazione vengono recuperate in circa 2 secondi.