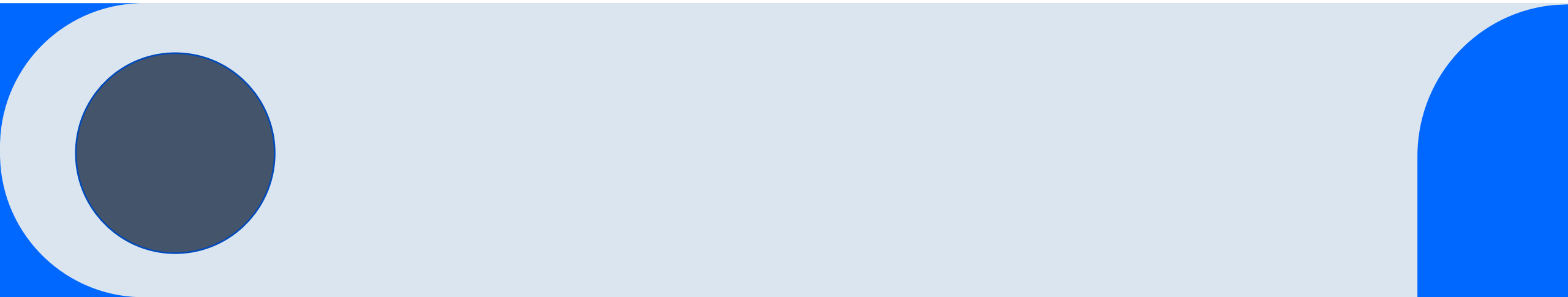


Introduzione alle LinkedList



Introduzione alle LinkedList

- Le LinkedList sono strutture dinamiche che consentono l'inserimento e la rimozione di elementi in modo efficiente rispetto agli array.
- Rispetto agli array, le LinkedList sono ideali per operazioni vicino all'inizio o alla fine della lista, offrendo maggiore flessibilità.
- Le LinkedList, conosciute anche come liste collegate, rappresentano un'importante risorsa per gestire dati in modo efficiente e dinamico.



Vantaggi delle LinkedList

- Le LinkedList sono flessibili e efficienti per le operazioni di inserimento e rimozione, risultando superiori agli array in questi scenari.
- Grazie alla crescita dinamica, le LinkedList possono adattarsi alle esigenze senza vincoli di dimensione fissa come negli array.
- L'utilizzo delle LinkedList è consigliato quando la manipolazione frequente della struttura dati richiede velocità e adattabilità.



Puntatore all'elemento successivo

- Ogni nodo di una LinkedList contiene un valore e un riferimento al nodo successivo, chiamato puntatore.
- Grazie al puntatore, è possibile collegare gli elementi della lista permettendo l'accesso sequenziale ai dati in modo efficiente.
- Le LinkedList consentono l'inserimento e la rimozione di elementi con maggiore efficienza rispetto agli array, specialmente in operazioni vicine all'inizio o alla fine della lista.



Esempio pratico: LinkedList di numeri

- La LinkedList è una struttura dati che collega nodi contenenti valori e riferimenti al nodo successivo, consentendo l'aggiunta e la rimozione efficiente di elementi.
- Nell'esempio con valori 5, 10 e 15, il primo nodo contiene 5 e punta al nodo con 10, che a sua volta punta al nodo con 15.
- Questo tipo di struttura è flessibile e ottimale per gestire dati dinamici e operazioni di inserimento o rimozione frequenti in programmazione.



Metodo di aggiunta in testa

- Creare un nuovo nodo con il valore da aggiungere e il riferimento al nodo attualmente in testa.
- Aggiornare il puntatore della testa della LinkedList in modo da puntare al nuovo nodo creato.
- L'operazione è veloce e efficiente poiché richiede solo l'aggiornamento del puntatore della testa.



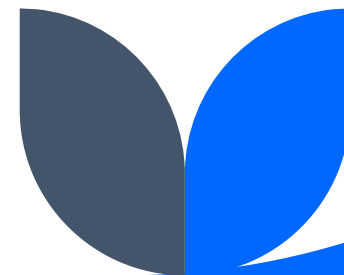
Metodo di rimozione in coda

- Modificare il puntatore dell'elemento precedente all'ultimo elemento affinché punti a NULL.
- Assicurarsi che il puntatore dell'ultimo elemento punti effettivamente a NULL per terminare correttamente la lista.
- Verificare la corretta gestione della memoria per evitare memory leaks o accessi non validi dopo la rimozione.



Metodo di ricerca

- Per cercare un elemento in una LinkedList, si inizia dalla testa e si scorrono i nodi confrontando i valori fino a trovare l'elemento o arrivare alla fine della lista.
- Questo processo richiede un'iterazione attraverso i nodi, confrontando i valori uno per uno fino a individuare l'elemento desiderato all'interno della struttura dati.
- La complessità temporale dipende dalla posizione dell'elemento nella lista: va da $O(1)$ per l'inizio/fine fino a $O(n)$ per l'elemento in una posizione casuale.



LinkedList vs Array

- Le LinkedList non richiedono dimensione fissa, rendendole più flessibili nella gestione dei dati rispetto agli array.
- Le operazioni di inserimento e rimozione sono più efficienti nelle LinkedList, specialmente con frequenti modifiche alla struttura dei dati.
- LinkedList è consigliata quando si lavora vicino all'inizio o alla fine della lista per ottenere prestazioni ottimali.



Utilizzo delle LinkedList

- Le LinkedList possono essere utilizzate per implementare liste, code e stack, offrendo flessibilità nella gestione di dati complessi.
- Grazie alla struttura dinamica, le LinkedList permettono l'inserimento e la rimozione efficiente di elementi in diverse posizioni della lista.
- La versatilità delle LinkedList le rende ideali per situazioni in cui le operazioni di aggiunta e rimozione sono frequenti o necessarie.



Conclusione

- Le LinkedList consentono l'aggiunta e la rimozione di elementi in modo dinamico, offrendo flessibilità e efficienza nella gestione dei dati.
- Comprendere i concetti fondamentali delle LinkedList è cruciale per lo sviluppo di competenze solide in programmazione e ottimizzazione delle prestazioni.
- Rispetto agli array, le LinkedList si distinguono per la capacità di gestire l'inserimento e la rimozione degli elementi in maniera più efficiente e versatile.

