

- Fusione ordinata di liste
 - Scrivere una funzione `merge_sorted_lists` che prende due liste ordinate in modo crescente e restituisce una nuova lista ordinata contenente tutti gli elementi di entrambe le liste. La funzione non deve modificare le liste originali.
- Rimozione di duplicati
 - Implementare una funzione `remove_duplicates` che elimina tutti i nodi duplicati da una lista, mantenendo solo la prima occorrenza di ogni valore. La lista non è necessariamente ordinata.
- Rotazione di una lista
 - Creare una funzione `rotate_list` che ruota una lista di k posizioni verso destra. Per esempio, data la lista 1->2->3->4->5 e k=2, il risultato dovrebbe essere 4->5->1->2->3.
- Palindromo di lista
 - Scrivere una funzione `is_palindrome` che determina se una lista è un palindromo. La funzione dovrebbe restituire 1 se la lista è un palindromo, 0 altrimenti.
- Intersezione di liste
 - Implementare una funzione `find_intersection` che trova il nodo di intersezione di due liste (se esiste). Le liste possono avere lunghezze diverse.
- Riordino alternato
 - Creare una funzione `reorder_alternate` che riordina una lista in modo che ogni altro elemento sia maggiore dei suoi vicini. Ad esempio, data la lista 1->2->3->4->5, il risultato potrebbe essere 1->3->2->5->4.
- Suddivisione di lista
 - Implementare una funzione `split_list` che divide una lista in due parti: una contenente i nodi con valori pari e l'altra con valori dispari, mantenendo l'ordine relativo originale.