**Linked List**

Zalety Linked List:

1. **Dynamiczna struktura danych**: Listy jednokierunkowe są dynamicznymi strukturami danych, co oznacza, że ​​mogą zmieniać swoją wielkość w trakcie działania programu, w przeciwieństwie do tablic, których rozmiar jest stały po utworzeniu.
2. **Efektywne dodawanie i usuwanie**: W przeciwieństwie do tablic, w których dodawanie lub usuwanie elementów może wymagać przesunięcia wszystkich elementów w tablicy, listy jednokierunkowe umożliwiają dodawanie i usuwanie elementów w stałym czasie O(1), jeśli dostępny jest wskaźnik na odpowiedni element.
3. **Łatwe sortowanie**: Listy jednokierunkowe ułatwiają sortowanie, ponieważ mogą być łatwo przesuwane i zmieniane bez konieczności przekładania elementów, co jest często potrzebne w przypadku tablic.
4. **Brak konieczności alokacji statycznej pamięci**: Listy jednokierunkowe nie wymagają alokacji statycznej pamięci, ponieważ nowe elementy mogą być dynamicznie alokowane w pamięci w miarę potrzeb.

Wady Linked List:

1. **Wolniejszy dostęp do elementów:** W przeciwieństwie do tablic, dostęp do elementów listy jednokierunkowej nie jest natychmiastowy i wymaga przejścia przez wszystkie poprzedzające elementy w celu dotarcia do pożądanego elementu. To może prowadzić do wolniejszego dostępu, szczególnie w przypadku dużej ilości danych.
2. **Pamięć zużywana na wskaźniki**: Każdy element w liście jednokierunkowej zawiera wskaźnik do następnego elementu, co zajmuje dodatkową pamięć w porównaniu do tablicy, gdzie wszystkie elementy znajdują się w jednym ciągłym obszarze pamięci.
3. **Brak wsparcia dla szybkiego dostępu losowego**: Lista jednokierunkowa nie wspiera szybkiego dostępu do elementów na podstawie ich indeksów, co oznacza, że aby uzyskać dostęp do konkretnego elementu, konieczne jest przejście przez listę od początku lub od danego wskaźnika.
4. **Potencjalna fragmentacja pamięci**: Dynamiczna alokacja pamięci podczas dodawania nowych elementów do listy może prowadzić do fragmentacji pamięci, co może prowadzić do marnowania pamięci i potencjalnie obniżać wydajność programu.

Mimo tych wad, listy jednokierunkowe nadal są używane w wielu przypadkach, szczególnie gdy wymagane są dynamiczne struktury danych, takie jak kolejki, stosy lub kiedy operacje dodawania i usuwania są bardziej krytyczne niż szybki dostęp do elementów.