

Лабораторна робота 7

ДОСЛІДЖЕННЯ ПОБУДОВИ ТА ВИКОРИСТАННЯ СТРУКТУР ДАНИХ

Мета лабораторної роботи – дослідити типи лінійних та нелінійних структур даних, навчитись користуватись бібліотечними реалізаціями структур даних та будувати власні.

Теоретичні відомості

Структура даних – це засіб зберігання даних для забезпечення доступу до них для читання та запису певним способом. Не існує однієї структури даних, що працює найкращим чином для всіх цілей, тому у різних умовах та обмеженнях використовуються різні структури.

Лінійні структури даних – це такі структури, елементи яких мають форму послідовності або лінійного списку. До лінійних структур даних відносять масиви, стеки, черги та зв'язані лінійні списки.

Стек – це структура даних, яка побудована на метафорі «магазин автомату» або «трубка з одним відкритим кінцем» та забезпечує доступ до елементів тільки з одного кінця за правилом «Перший увійшов – останній вийшов» - First in – Last out (LIFO) (рис.1). Стек забезпечує дві операції з його елементами - «помістити» - Push, та «вийняти» - Pop. Один й той же елемент може бути витягнутий зі стеку тільки один раз, коли він знаходиться на вершині.

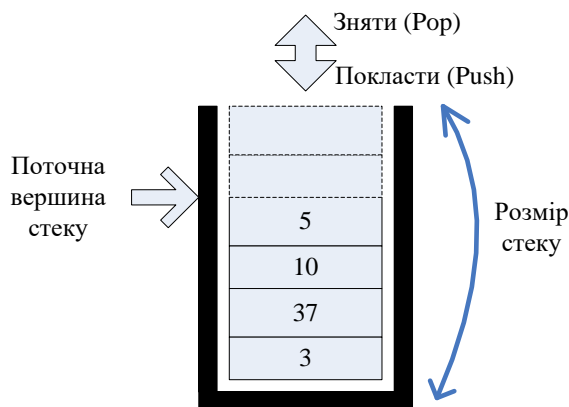


Рис 1. Стек

Черга – це структура даних, яка побудована на метафорі «черга обслуговування» або «однонаправлена трубка з двома відкритими кінцями» та забезпечує правило «Перший увійшов – перший вийшов» - First in – First out (FIFO) (рис.2). Черга забезпечує дві операції з його елементами – «Помістити у чергу» - Enqueue та «Взяти з черги» - Dequeue. Елемент може бути взятий з черги тільки один раз.

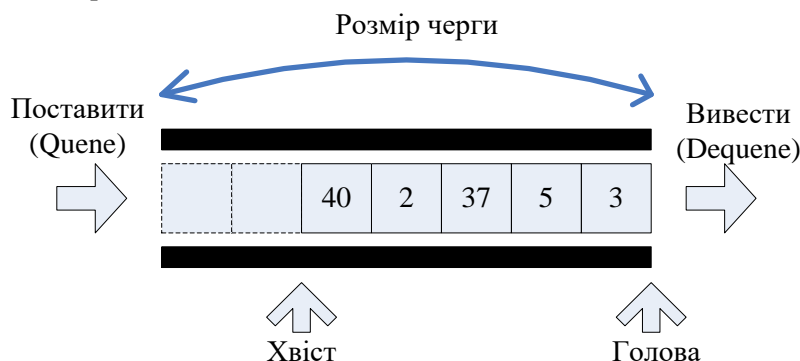


Рис. 2 Черга

Зв'язаний лінійний список – це структура, всі елементи якої зв'язані один з іншим за допомогою посилань (рис.3). На перший елемент списку посилається окрема змінна, яка містить його адресу. Кожний елемент списку містить показник з адресою наступного елементу. Останній елемент списку містить посилання на *null*.

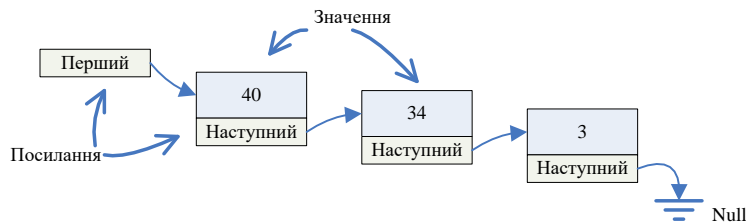


Рис. 3 Зв'язаний список

Нелінійні структури даних – це структури, елементи яких не можливо представити у вигляді послідовності. Дерево – це нелінійна структура даних, яка зберігає їх у зв'язаних вузлах, що утворюють ієрархію (рис.4). Кожний вузол дерева (окрім кореневого) зв'язаний зі старшим вузлом та може бути зв'язаним з молодшими вузлами. Бінарне дерево – це таке дерево, кожний вузол якого може мати тільки два молодших вузла.

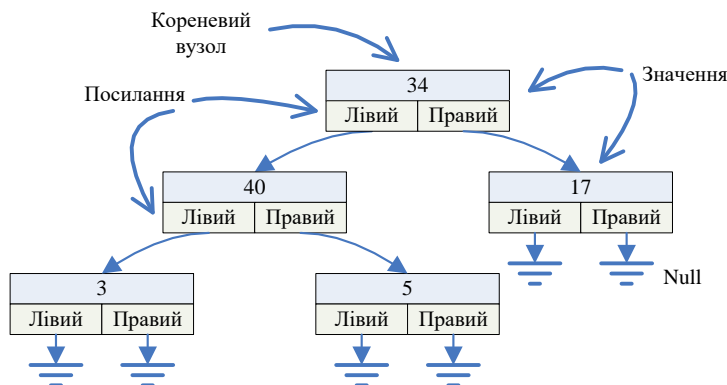


Рис. 4 Дерево

Завдання

Написати програми мовами C++ та C#, у яких описати власну структуру даних згідно з варіантом. Продемонструвати функціональність розробленої структури шляхом застосування всіх її операцій (створити декілька об'єктів структур, додати в них певну кількість даних, викликати операції над даними, видалити певну кількість даних, тощо):

Таблиця 1

№	Тип даних елементів списку	Тип списку	Спосіб додавання елементу списку	Операції зі списком
1	char	Односпрямований	Включення до початку	Знайти перше входження символу «!» Видалити всі символи «а»
2	float	Односпрямований	Включення в кінець списку	Знайти кількість елементів більших за 10.5 Видалити елементи менші за 2.6
3	short	Односпрямований	Включення до початку	Знайти кількість елементів, кратних 3. Видалити елементи більше за середнє значення.
4	int	Односпрямований	Включення після першого елементу	Знайти кількість елементів, кратних 5. Видалити елементи, які розташовані після максимального елементу.
5	double	Односпрямований	Включення в кінець списку	Знайти кількість елементів більших за середнє значення. Видалити елементи, що розташовані на парних позиціях.
6	short	Односпрямований	Включення в кінець списку	Знайти кількість елементів, кратних 4. Замінити елементи, що розташовані на парних позиціях, на «0».
7	float	Односпрямований	Включення після	Знайти кількість елементів, більших за 3.14

№	Тип даних елементів списку	Тип списку	Спосіб додавання елементу списку	Операції зі списком
			другого елементу	Видалити елементи, які більші за середнє значення.
8	char	Двоспрямований	Включення в кінець списку	Знайти кількість елементів «№», які розташовані на парних позиціях списку Видалити символи більші за «а»
9	double	Двоспрямований	Включення до початку	Знайти кількість елементів менших за середнє значення Видалити елементи, які розташовані після максимального елементу
10	long	Двоспрямований	Включення в кінець списку	Знайти кількість елементів, кратних 5, що розташовані на парних позиціях. Видалити елементи більші за середнє значення
11	float	Односпрямований	Включення до початку	Знайти кількість елементів, більших за середнє значення. Видалити всі від'ємні елементи.
12	int	Односпрямований	Включення після першого елементу	Знайти кількість елементів, кратних 2, які розташовані на парних позиціях. Видалити елементи, які розташовані після мінімального елементу
13	char	Односпрямований	Включення в кінець списку	Знайти кількість входжень символу «*» Видалити символи, які розташовані після «#»
14	double	Двоспрямований	Включення до початку	Знайти кількість елементів менших за середнє значення Видалити елементи, які розташовані до максимального елементу.
15	short	Односпрямований	Включення в кінець списку	Знайти кількість елементів, кратних 7 Замінити елементи більші за середнє значення на «0»

Контрольні запитання

1. Що таке структура даних?
2. Наведіть типи структур даних.
3. Поясніть, чим відрізняється список від словника
4. Поясніть, чим відрізняється стек від черги
5. Поясніть сутність побудови дерева
6. Поясніть сутність векторної реалізації структур даних
7. Поясніть сутність зв'язаної реалізації структур даних
8. Наведіть приклад опису типа вузла лінійного списку у зв'язаній реалізації та поясніть устрій вузла
9. Наведіть приклад опису односпрямованого та двоспрямованого списків та поясніть різницю.
10. Наведіть приклади реалізації структур даних у C#
11. Наведіть приклади реалізації структур даних у C++