Лекция №19
по дисциплине
«ТЕОРИЯ АЛГОРИТМОВ»

СВЯЗАННЫЙ СПИСОК

Преподаватель:
Золотоверх Д.О.

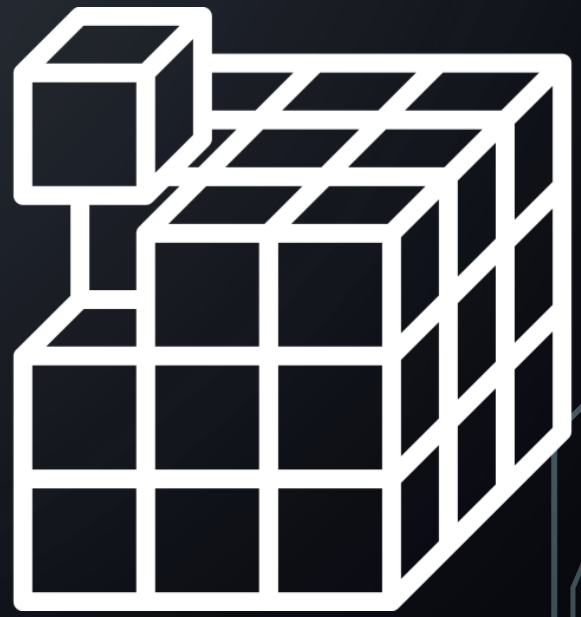
СТРУКТУРА ДАННЫХ

Способ организации информации: ее формат хранения, способы изменения и доступ.

Такой информацией может быть совокупность и однотипных, и многотипных данных.

Данные могут быть связаны.

Управление данными предоставлено определенным способом.



СТРУКТУРА ДАННЫХ

- имеет внутреннюю форму, данные могут быть связаны;
- может иметь несколько разных типов данных;
- хранит информацию об предмете;
- значение нельзя изменить прямым способом, только с помощью специальной операции;
- нужно учитывать проблему сложности вычислений.

ПРИМЕРЫ СТРУКТУР ДАННЫХ

Примеры структур данных:

- Стек (определенный порядок доступа и модификации);
- Очередь (схож в реализации со Стеком);
- Связанный список (массив, но элементы связаны опр. способом);
- Множество (может хранить только уникальные элементы);
- Хеш-таблица (пара ключ-значение);
- Дерево (древовидная структура связанных элементов);
- Граф (связанное множество).

АБСТРАКЦИЯ

Структуры данных являются более абстрактными сущностями, чем массивы и типы данных.

Они определяются, прежде всего, своим интерфейсом: набором разрешенных операций, которые могут выполняться с ними.

Интерфейс этих структур проектируется с расчетом на поддержку ограничений доступа.

Базовый механизм, используемый для их реализации, обычно остается невидимым для пользователя.

МАССИВ

Массив не является структурой данных (модификация может быть выполнена только напрямую).

Он может состоять из одного типа данных (целочисленный int, булевой bool, дробный float, символьный char и тд.)

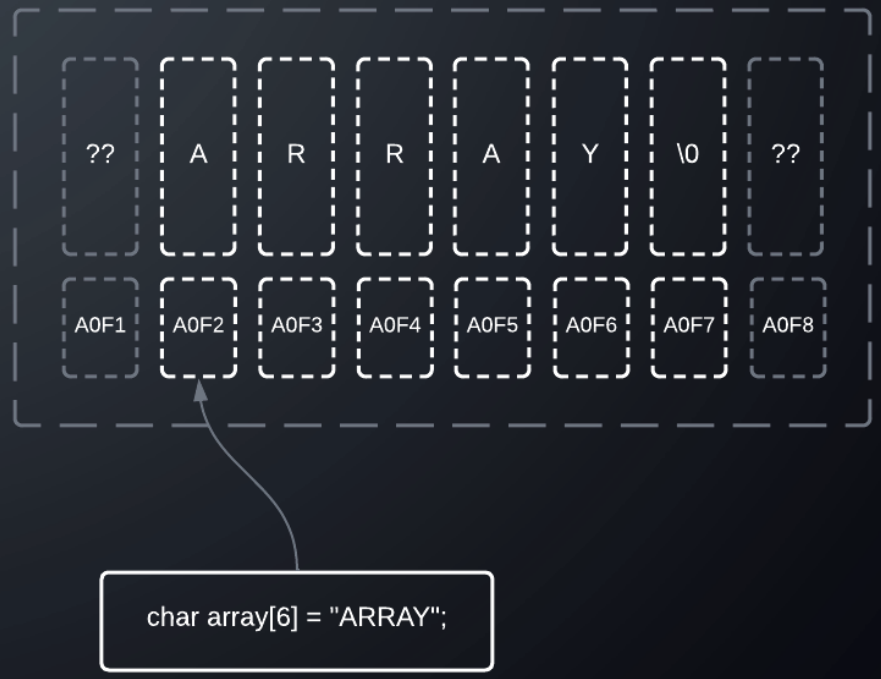
Он фиксированного размера, должен быть определен заранее и не может быть изменен на лету.

УСТРОЙСТВО

Массив состоит из набора однотипных данных. Значения хранятся последовательно, могут быть индексированы по индексу (начинается с нуля).

Последовательность хранения значений означает, что информация хранится друг за другом в одном «куске» памяти.

По этому нет возможности изменить его размер – нужно создавать новый массив другого размера.



СИНТАКСИС

Объявление массива может состоять из четырех частей:

- указание типа;
- название переменной;
- указание длины (не обязательно);
- перечень стартовых значений (не обязательно);

Синтаксис:

```
int myInts[6];
int myPins[] = {2, 4, 8, 3, 6};
int myVals[6] = {2, 4, -8, 3, 2};
char message[6] = "hello";
```

```
#include <iostream>
using namespace std;


int main() {
    int array[] = {3, 2, 1};
    for (int i = 0; i < 3; i++)
{
        printf("%d\n",
        array[i]);
    }
}
```

ФИКСИРОВАННЫЙ РАЗМЕР

Если все таки нужно изменить размер массива, нет другого выбора, как создать новый с нужным размером.

При создании нового также необходимо скопировать содержимое старого.

Такая операция не есть бесплатной, помимо дополнительной памяти, что требуется для нового массива, необходимо еще и перебрать все элементы, таким образом имеем сложность:

$O = n$

```
int main() {
    int oldLen = 5;
    int oldArray[] = {1, 2, 3, 4, 5};

    int newValue = 6;

    int newLen = 8;
    int newArray[newLen];

    for (int i = 0; i < oldLen; i++) {
        newArray[i] = oldArray[i];
    }

    newArray[5] = newValue;
}
```

СВЯЗАННЫЙ СПИСОК

Является структурой данных, которая состоит из упорядоченного набора значений.

Все элементы списка «связаны» между собой:

- имеют значение;
- элемент ссылку на следующий (предыдущий элемент);

Главным преимуществом является гибкость динамический размер, его можно менять во время выполнения программы.

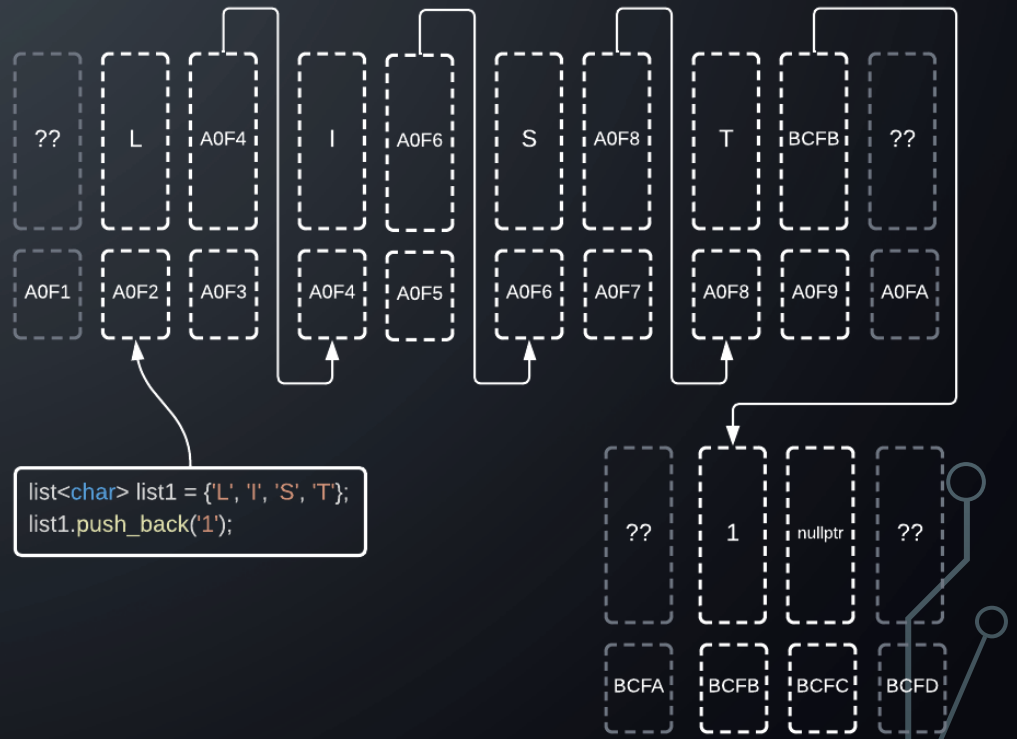
УСТРОЙСТВО ЛИНЕЙНОГО СВЯЗАННОГО СПИСКА

Список состоит из элементов одного типа, связанных между собой последовательно посредством указателей.

Каждый элемент списка имеет указатель на следующий элемент.

Последний элемент списка указывает на NULL.

Элемент, на который нет указателя, является первым (головным) элементом списка.

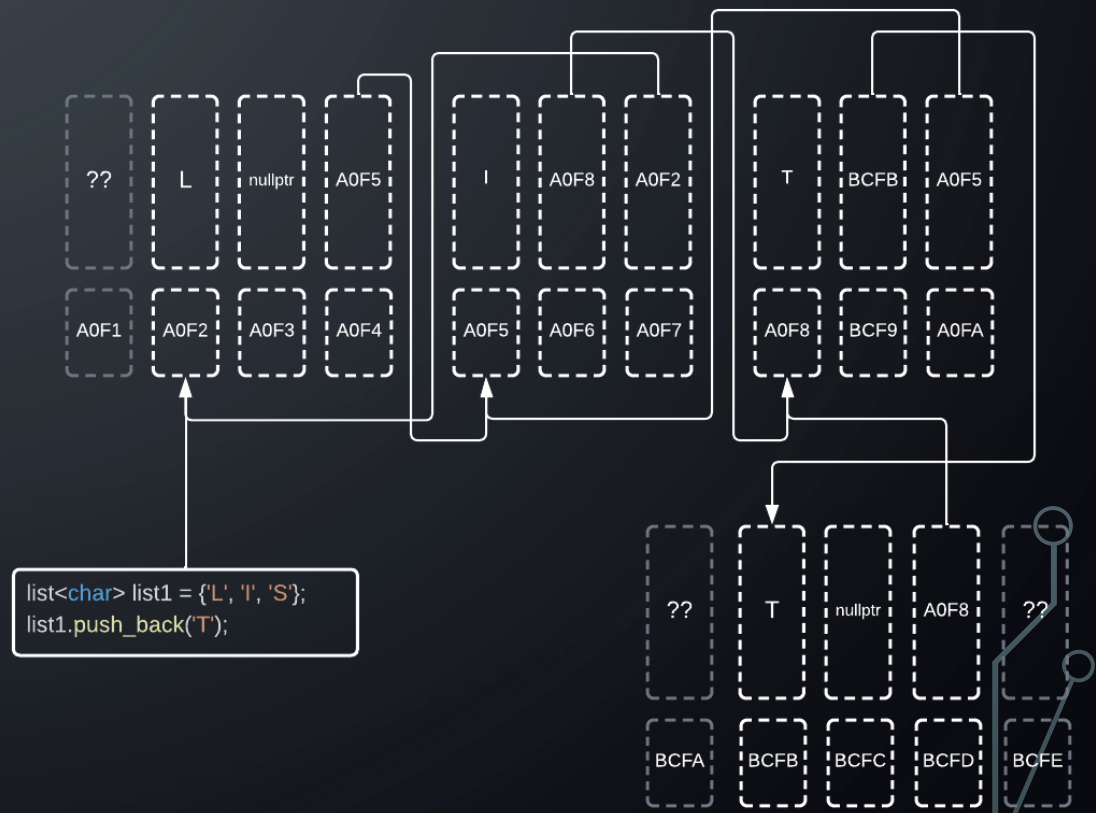


УСТРОЙСТВО ДВУСВЯЗНЫЙ СПИСОК

Устройство схожее с линейным связанным, за исключением только того, что каждый элемент имеет указатель не только на следующий, а и на предыдущий элементы.

Как и односвязный список, двусвязный допускает только последовательный доступ к элементам, но при этом дает возможность перемещения в обе стороны.

Именно такой механизм имеет список из стандартной библиотеки std:list.



СИНТАКСИС

Объявление стека состоит из трех частей:

- указание типа шаблона;
- указание типа данных;
- название переменной.

Списко можно импортировать с помощью директивы #include <list>

Синтаксис:

```
список <тип_данных> названиеСписка;
названиеСписка.метод();
```

```
#include <iostream>
#include <list>

using namespace std;


int main() {
    list<char> list1;
    list1.push_back('L');
    list1.push_back('I');
    list1.push_back('S');
    list1.push_back('T');

    for (char c : list1) {
        printf("%c\t", c);
    }
}
```

МЕТОДЫ РАБОТЫ СО СПИСКОМ

Неполный перечень методов списка

Название метода	Описание метода	Аргументы	Возврат
front()	Смотрит первый элемент списка	Ничего	Элемент списка
back()	Смотрит последний элемент списка	Ничего	Элемент списка
push_front()	Добавляет элемент в начало списка	Новый элемент	Ничего
pop_front()	Убирает элемент из начала списка	Ничего	Ничего
remove()	Удаляет элементы с указанным значением	Элемент списка	Ничего
reverse()	«переворачивает список»	Ничего	Ничего
sort()	сортирует список	Ничего	Ничего

ПРЕИМУЩЕСТВА СВЯЗАННОГО СПИСКА

В сравнении с массивами, связанные списки имеют следующие преимущества:

- эффективное (за константное время O = k) добавление и удаление элементов. Не нужно создавать новый список и копировать все содержимое старого;
- размер ограничен только объёмом памяти компьютера и разрядностью указателей.
- динамическое добавление и удаление элементов.

НЕДОСТАТКИ СВЯЗАННОГО СПИСКА

Но существуют следующие недостатки:

- сложность прямого доступа к элементу, а именно определения физического адреса по его индексу (порядковому номеру) в списке.
- на поля-указатели (указатели на следующий и предыдущий элемент) расходуется дополнительная память (в массивах, указатели не нужны)
- некоторые операции со списками медленнее, чем с массивами, так как к произвольному элементу списка можно обратиться, только пройдя все предшествующие ему элементы;

СПАСИБО ЗА ВНИМАНИЕ!