Динамическое расширение массива

Урок №26

Разбор домашней работы

Повторение

Повторение

Тип массива	Статический	Динамический
Расположение	?	
Время жизни		
Создание		
Удаление		

Повторение

Тип массива	Статический	Динамический
Расположение	Stack (стек)	?
Время жизни		
Создание		
Удаление		

Повторение

Тип массива	Статический	Динамический
Расположение	Stack (стек)	Heap (куча)
Время жизни	?	
Создание		
Удаление		

Повторение

Тип массива	Статический	Динамический
Расположение	Stack (стек)	Heap (куча)
Время жизни	Нахождение в области видимости	?
Создание		
Удаление		

Повторение

Тип массива	Статический	Динамический
Расположение	Stack (стек)	Heap (куча)
Время жизни	Нахождение в области видимости	Явно, от создания до удаления
Создание	?	
Удаление		

Повторение

Тип массива	Статический	Динамический
Расположение	Stack (стек)	Heap (куча)
Время жизни	Нахождение в области видимости	Явно, от создания до удаления
Создание	Объявление	?
Удаление		

Повторение

Тип массива	Статический	Динамический
Расположение	Stack (стек)	Heap (куча)
Время жизни	Нахождение в области видимости	Явно, от создания до удаления
Создание	Объявление	Вызов оператора new[]
Удаление	?	

Повторение

Тип массива	Статический	Динамический
Расположение	Stack (стек)	Heap (куча)
Время жизни	Нахождение в области видимости	Явно, от создания до удаления
Создание	Объявление	Вызов оператора new[]
Удаление	Выход из области видимости	?

Повторение

Тип массива	Статический	Динамический
Расположение	Stack (стек)	Heap (куча)
Время жизни	Нахождение в области видимости	Явно, от создания до удаления
Создание	Объявление	Вызов оператора new[]
Удаление	Выход из области видимости	Вызов оператора delete[]

Повторение

Тип массива	Статический	Динамический
Создание	`int arr[5];`	`int* arr = new int[n];`
Считывание данных	`cin >> arr[i];`	
Вывод данных	`cout << arr[i];`	
Удаление	Выход из области видимости	`delete[] arr;`

Повторение

Почему константа?

Тип массива	Статический	Динамический
Создание	`int arr[5];`	`int* arr = new int[n];`
Считывание данных	`cin >> arr[i];`	
Вывод данных	`cout << arr[i];`	
Удаление	Выход из области видимости	`delete[] arr;`

Повторение

Почему константа?

Тип массива	Статический	Динамический
Создание	`int arr[5];`	`int* arr = new int[n];`
Считывание данных	`cin >> arr[i];`	
Вывод данных	`cout << arr[i];`	
Удаление	Выход из области видимости	`delete[] arr;`

Когда происходит?

Теория

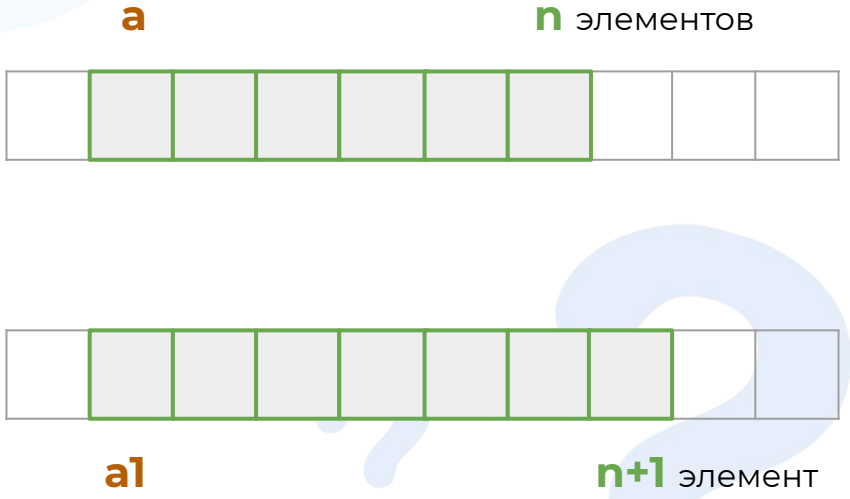
20 минут

Добавление элемента в динамический массив

Добавление 1 элемента

Добавить элемент **x** на позицию **k** в массиве **a** из **n** элементов

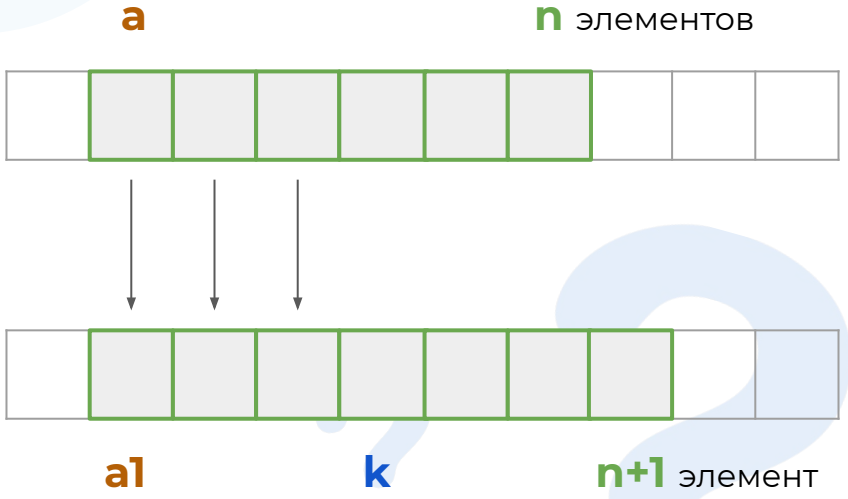
1. Создаем новый массив **a1** на **n+1** элемент



Добавление 1 элемента

Добавить элемент **x** на позицию **k** в массиве **a** из **n** элементов

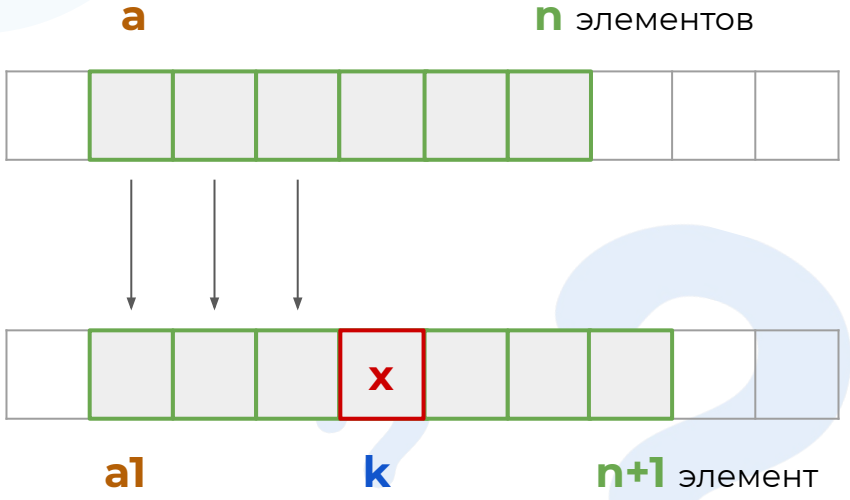
1. Создаем новый массив **a1** на **n+1** элемент
2. Копируем элементы из массива **a** в массив **a1** от индекса 0 до **k**



Добавление 1 элемента

Добавить элемент **x** на позицию **k** в массиве **a** из **n** элементов

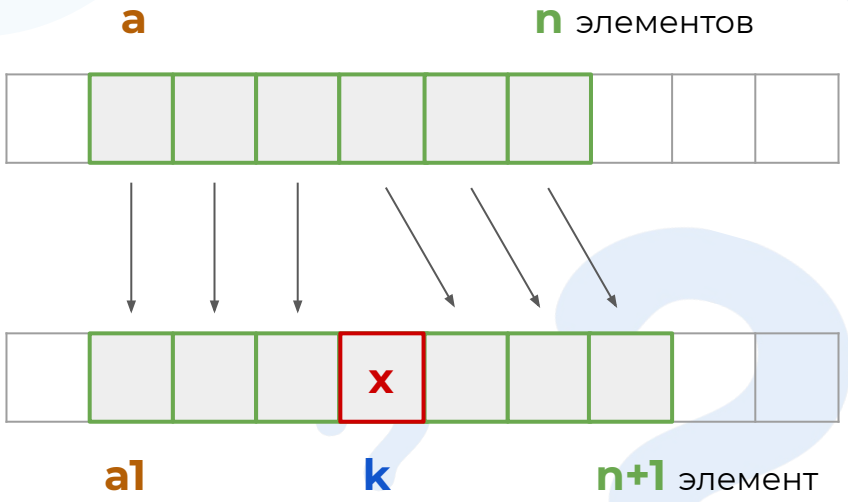
1. Создаем новый массив **a1** на **n+1** элемент
2. Копируем элементы из массива **a** в массив **a1** от индекса 0 до **k**
3. Присваиваем по индексу **k** в массиве **a1** значение **x**



Добавление 1 элемента

Добавить элемент **x** на позицию **k** в массиве **a** из **n** элементов

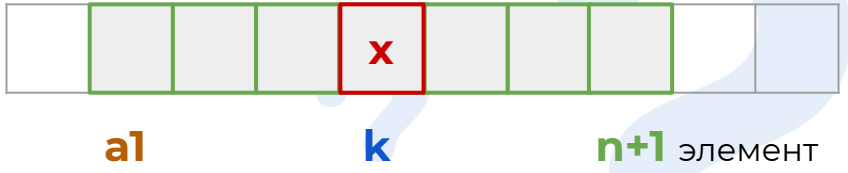
1. Создаем новый массив **a1** на **n+1** элемент
2. Копируем элементы из массива **a** в массив **a1** от индекса 0 до **k**
3. Присваиваем по индексу **k** в массиве **a1** значение **x**
4. Копируем элементы из массива **a** в массив **a1** от индекса **k** до индекса **n**



Добавление 1 элемента

Добавить элемент **x** на позицию **k** в массиве **a** из **n** элементов

1. Создаем новый массив **a1** на **n+1** элемент
2. Копируем элементы из массива **a** в массив **a1** от индекса 0 до **k**
3. Присваиваем по индексу **k** в массиве **a1** значение **x**
4. Копируем элементы из массива **a** в массив **a1** от индекса **k** до индекса **n**
5. Удаляем массив **a** (освобождаем память)



Добавление 1 элемента

Добавить элемент **x** на позицию **k** в массиве **a** из **n** элементов

```
int* add(int* a, int* n, int k, int x){
    int* a1 = new int[*n + 1];
    for (int i = 0; i < k; i++)
        a1[i] = a[i];
    a1[k] = x;
    for (int i = k + 1; i <= *n; i++)
        a1[i] = a[i - 1];
    delete[] a;
    *n += 1;
    return a1;
}
```

Теперь напишем этот алгоритм на C++!

Добавление 1 элемента

Добавить элемент **x** на позицию **k** в массиве **a** из **n** элементов

```
int* add(int* a, int* n, int k, int x){
    int* a1 = new int[*n + 1];
    for (int i = 0; i < k; i++)
        a1[i] = a[i];
    a1[k] = x;
    for (int i = k + 1; i <= *n; i++)
        a1[i] = a[i - 1];
    delete[] a;
    *n += 1;
    return a1;
}
```

Не забываем увеличить переменную размера массива!

Добавление 1 элемента

Добавить элемент **x** на позицию **k** в массиве **a** из **n** элементов

```
int* add(int* a, int* n, int k, int x)

int main() {
    // ввод
    int* a = new int[n];
    // ввод
    a = add(a, &n, k, x);
    // вывод
    delete[] a;
}
```

*Пример использования алгоритма в **main***

Практика

20 минут

Теория

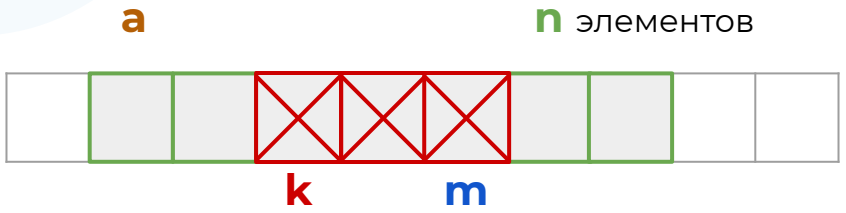
15 минут

Удаление элементов из динамического массива

Удаление элементов

Удалить элементы с индекса **k** по **m** в массиве **a** из **n** элементов

1. Вычисляем количество удаляемых элементов **count = m - k + 1**

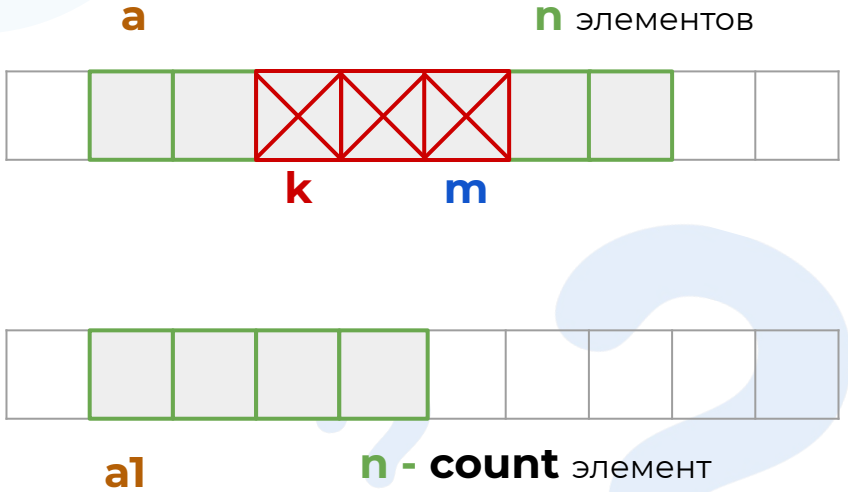


Удаление элементов

Удалить элементы с индекса **k** по **m** в массиве **a** из **n** элементов

1. Вычисляем количество удаляемых элементов **count = m - k + 1**

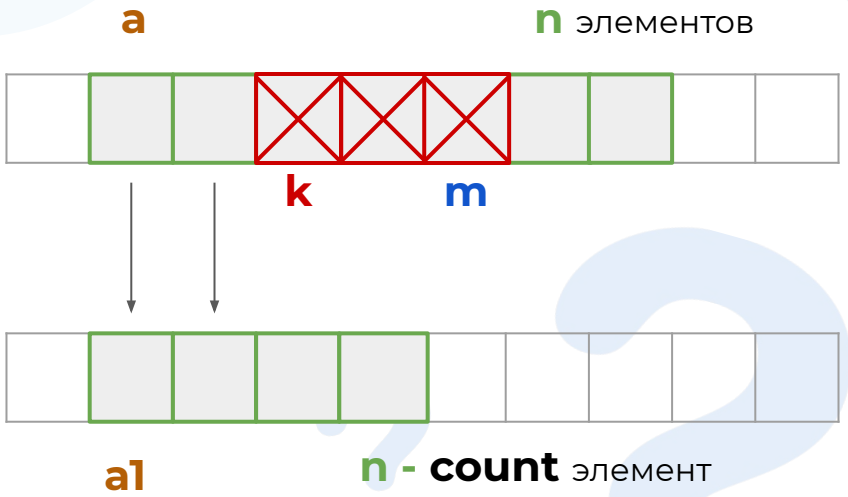
2. Создаем новый массив **a1** на **n** - **count** элементов



Удаление элементов

Удалить элементы с индекса **k** по **m** в массиве **a** из **n** элементов

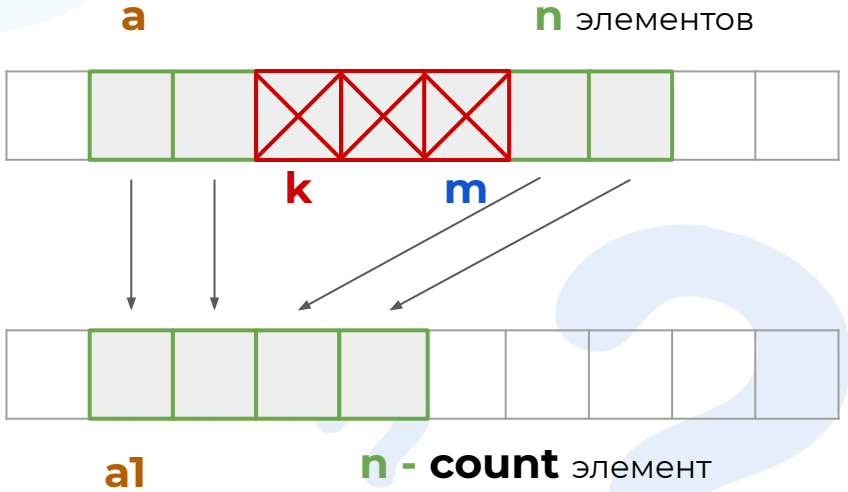
1. Вычисляем количество удаляемых элементов **count = m - k + 1**
2. Создаем новый массив **a1** на **n** - **count** элементов
3. Копируем элементы из массива **a** в массив **a1** от индекса 0 до индекса k.



Удаление элементов

Удалить элементы с индекса **k** по **m** в массиве **a** из **n** элементов

1. Вычисляем количество удаляемых элементов **count = m - k + 1**
2. Создаем новый массив **a1** на **n** - **count** элементов
3. Копируем элементы из массива **a** в массив **a1** от индекса 0 до индекса k.
4. Копируем элементы из массива **a** в массив **a1** от индекса **m + 1** до индекса **n**.



Удаление элементов

Удалить элементы с индекса **k** по **m** в массиве **a** из **n** элементов

1. Вычисляем количество удаляемых элементов **count = m - k + 1**
2. Создаем новый массив **a1** на **n** - **count** элементов
3. Копируем элементы из массива **a** в массив **a1** от индекса 0 до индекса k.
4. Копируем элементы из массива **a** в массив **a1** от индекса **m + 1** до индекса **n**.
5. Удаляем массив **a1** (освобождаем память)

a1 **n - count** элемент

Удаление элементов

Удалить элементы с индекса **k** по **m** в массиве **a** из **n** элементов

```
int* del(int* a, int* n, int k, int m) {
    int cnt = m - k + 1;
    int* a1 = new int[*n - cnt];
    for (int i = 0; i < k; i++)
        a1[i] = a[i];
    for (int i = m + 1; i < *n; i++)
        a1[i - cnt] = a[i];
    delete[] a;
    *n -= cnt;
    return a1;
}
```

Теперь напишем этот алгоритм на C++!

Удаление элементов

Удалить элементы с индекса **k** по **m** в массиве **a** из **n** элементов

```
int* del(int* a, int* n, int k, int m) {
    int cnt = m - k + 1;
    int* a1 = new int[*n - cnt];
    for (int i = 0; i < k; i++)
        a1[i] = a[i];
    for (int i = m + 1; i < *n; i++)
        a1[i - cnt] = a[i];
    delete[] a;
    *n -= cnt;
    return a1;
}
```

Не забываем уменьшить переменную размера массива!

Удаление элементов

Удалить элементы с индекса **k** по **m** в массиве **a** из **n** элементов

```
int* del(int* a, int* n, int k, int m);

int main() {
    //ввод...
    int* a = new int[n];
    //ввод...
    a = del(a, &n, k, m);
    //вывод...
    delete[] a;
}
```

*Пример использования алгоритма в **main***

Удаление элементов

Удалить элементы с индекса **k** по **m** в массиве **a** из **n** элементов

```
int* del(int* a, int* n, int k, int m);

int main() {
    //ввод...
    int* a = new int[n];
    //ввод...
    a = del(a, &n, k, m);
    //вывод...
    delete[] a;
}
```

Как называется такое объявление функции?

Практика

25 минут

Итоги урока

1) Как создать динамический массив?
2) Как добавить новый элемент в динамический массив?
3) Как удалить элементы из динамического массива?
4) Почему нельзя просто уменьшить или увеличить размер динамического массива?