НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Алгоритмизация и программирование

Программирование на C/C++

(ч.8 – сортировка массивов)

Беркунский Е.Ю., кафедра ИУСТ, НУК
eugeny.berkunsky@gmail.com
http://berkut.homelinux.com

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Сортировка массивов

Формулировка задачи:

- Дан массив A из n элементов:
  a[0], a[1], a[2],… a[n-1]
- Требуется переставить элементы этого массива так, чтобы после перестановки они были упорядочены по неубыванию:
  $a[0] \le a[1] \le a[2] \le \ldots \le a[n-1]$

Примечание. Хотя, в нашем случае, массивы имеют не очень большой размер, будем считать, что заводить вспомогательный массив недопустимо – расходуется “лишняя память”

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Сортировка массивов

Алгоритмы сортировки:

- простые
- усовершенствованные

Мы рассмотрим простые методы:

- Сортировка выбором
- Сортировка обменами
- Сортировка вставками

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Сортировка выбором

Шаги алгоритма:

- находим номер минимального значения в массиве
- производим обмен этого значения со значением первой неотсортированной позиции (обмен не нужен, если минимальный элемент уже находится на данной позиции)
- теперь сортируем хвост массива, исключив из рассмотрения уже отсортированные элементы

Сортировка массивов

```
#include <iostream>

using namespace std;

void chooseSort(int a[], int size);

void printArr(int a[], int size)
{
    for(int i=0; i<size; i++) {
        cout << a[i] << " ";
    }
    cout << "\n";
}

int main()
{
    int arr[] = {1, -5, 7, -2, 0, 11, 6, 8, -3, -4};
    chooseSort(arr, 10);
    printArr(arr, 10);
    return 0;
}
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Сортировка выбором

```
void chooseSort(int a[], int size)
{
    for (int i=0; i<size-1; i++) {
        int nmin = i;
        for (int j=i+1; j<size; j++){
            if (a[j]<a[nmin]) nmin = j;
        }
        if (i != nmin) {
            int t = a[nmin];
            a[nmin] = a[i];
            a[i] = t;
        }
    }
}
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Сортировка выбором

Демонстрация

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Сортировка обменами

Шаги алгоритма:

- Алгоритм состоит из повторяющихся проходов по сортируемому массиву.
- За каждый проход элементы последовательно сравниваются попарно и, если порядок в паре неверный, выполняется обмен элементов.
- Проходы по массиву повторяются N-1 раз или до тех пор, пока на очередном проходе не окажется, что обмены больше не нужны, что означает — массив отсортирован.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Сортировка массивов

```
#include <iostream>

using namespace std;

void bubbleSort(int a[], int size);

void printArr(int a[], int size)
{
    for(int i=0; i<size; i++) {
        cout << a[i] << " ";
    }
    cout << "\n";
}

int main()
{
    int arr[] = {1, -5, 7, -2, 0, 11, 6, 8, -3, -4};
    bubbleSort(arr, 10);
    printArr(arr, 10);
    return 0;
}
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Сортировка обменами

```
void bubbleSort(int a[], int size)
{
    for (int i=size-1; i>0; i--) {
        bool swap = false;
        for (int j=0; j<i; j++){
            if (a[j] > a[j+1]) {
                int t = a[j];
                a[j] = a[j+1];
                a[j+1] = t;
                swap = true;
            }
        }
        if (!swap) break;
    }
}
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Сортировка обменами

Демонстрация

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Сортировка вставками

Шаги алгоритма:

- Выбираем один из элементов массива и вставляем его на нужную позицию в уже отсортированной части массива
- Повторяем до тех пор, пока не достигнут конец массива.
- Метод выбора очередного элемента из исходного массива произволен; может использоваться практически любой алгоритм выбора. Обычно, элементы вставляются по порядку их появления во входном массиве.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Сортировка массивов

```
#include <iostream>

using namespace std;

void insertSort(int a[], int size);

void printArr(int a[], int size)
{
    for(int i=0; i<size; i++) {
        cout << a[i] << " ";
    }
    cout << "\n";
}

int main()
{
    int arr[] = {1, -5, 7, -2, 0, 11, 6, 8, -3, -4};
    insertSort(arr, 10);
    printArr(arr, 10);
    return 0;
}
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Сортировка вставками

```
void insertSort(int a[], int size)
{
    for (int i=1; i<size; i++)
    {
        int curr = a[i];
        int j = i-1;
        while (j>=0 && a[j]>curr) {
            a[j+1]=a[j];
            j--;
        }
        a[j+1] = curr;
    }
}
```

Сортировка вставками

Демонстрация

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

«Быстрая» сортировка QuickSort

Шаги алгоритма:

- из массива выбирается некоторый опорный элемент a[i],
- запускается процедура разделения массива, которая перемещает все ключи, меньшие, либо равные a[i], влево от него, а все ключи, большие, либо равные a[i] - вправо,
- теперь массив состоит из двух подмножеств, причем левое меньше, либо равно правого,
- для обоих подмассивов: если в подмассиве более двух элементов, рекурсивно запускаем для него ту же процедуру.
- В конце получится полностью отсортированная последовательность.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

«Быстрая» сортировка QuickSort

Псевдокод:

```
quickSort (массив a, нижняя_граница, верхняя_граница)
{
    Выбрать опорный элемент p - середину массива
    Разделить массив по этому элементу
    Если подмассив слева от p содержит более одного
    элемента,
        вызвать quickSort для него.
    Если подмассив справа от p содержит более одного
    элемента,
         вызвать quickSort для него.
}
```

QuickSort (1/2)

Как и с предыдущими сортировками, функция получает в качестве параметров массив и его размер

```
void sort(int arr[], int size)
{
    sort(arr, 0, size);
}
```

Опишем теперь основную часть функции

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

QuickSort (2/2)

```
void sort(int * a, int left, int right) {
// На входе - массив a[] и пределы сортировки
  int i = left, j = right-1;
  // поставить указатели на исходные места
  int temp, p;
  p = a[ (i + j) / 2 ];        // центральный элемент
  // процедура разделения
  do {
    while ( a[i] < p ) i++;
    while ( a[j] > p ) j--;

    if (i <= j) {
      temp = a[i]; a[i] = a[j]; a[j] = temp;
      i++; j--;
    }
  } while ( i<=j );
  // рекурсивные вызовы, если есть, что сортировать
  if ( j > left ) sort(a, left, j+1);
  if ( i < right-1 ) sort(a, i, right);
}
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

«Быстрая» сортировка QuickSort

Демонстрация

Спасибо!

Вопросы?

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Алгоритмизация и программирование

Программирование на C/C++

(ч.8 – сортировка массивов)

Беркунский Е.Ю., кафедра ИУСТ, НУК
eugeny.berkunsky@gmail.com
http://berkut.homelinux.com