1.5 Разработка кода программных модулей программного продукта

Разработка кода для программного продукта, который сортирует массивы данных четырьмя способами (пузырьковая сортировка, сортировка вставками сортировка Шелла, и быстрая сортировка), согласно техническому заданию, требует языка программирования, такого как C++. Ниже предоставлены коды программных модулей программного продукта.

Сортировка пузырьком

void pyzirok(int list[], int listLength)

{

while(listLength--)

{

bool swapped = false;

for(int i = 0; i < listLength; i++)

{

if(list[i] > list[i + 1])

{

swap(list[i], list[i + 1]);

swapped = true;

}

}

}

При пузырьковой сортировке сравниваются соседние элементы и меняются местами, если следующий элемент меньше предыдущего. Требуется несколько проходов по данным. Во время первого прохода сраваются первые два элемента в массиве. Если они не в порядке, они меняются местами и затем сравнивается элементы в следующей паре. При том же условии они так же меняются местами. Таким образом сортировка происходит в каждом цикле пока не будет достигнут конец массива.

Сортировка методом Вставки

void vstavka(int list[], int listLength)

{

for(int i = 1; i < listLength; i++)

{

int j = i - 1;

while(j >= 0 && list[j] > list[j + 1])

{

swap(list[j], list[j + 1]);

cout<<"\ndid";

j--;

}

}

}

При сортировке вставками массив разбивается на две области: упорядоченную и и неупорядоченную. Изначально весь массив является неупорядоченной областью. При первом проходе первый элемент из неупорядоченной области изымается и помещается в правильном положении в упорядоченной области. На каждом проходе размер упорядоченной области возрастает на 1, а размер неупорядоченной области сокращается на 1.

Сортировка методом Шелла

void shella(int list[], int listLength)

{

for(int step = listLength/2; step > 0; step /= 2)

{

for (int i = step; i < listLength; i += 1)

{

int j = i;

while(j >= step && list[j - step] > list[i])

{

swap(list[j], list[j - step]);

j-=step;

cout<<"\ndid";

}

}

}

Алгоритм включает в себя сортировку вставками. Исходный массив размером N разбивается на подмассивы с шагомN/2. Подмассивы сортируются вставками. Затем вновь разбиваются, но уже с шагом равным N/4. Цикл повторяется. Производим целочисленное деление шага на два каждую итерацию. Когда шаг становится равен 1, массив просто сортируется вставками.

Быстрая сортировка

void quickSort(int list[], int start, int end)

{

if(start < end)

{

int pivot = partition(list, start, end);

quickSort(list, start, pivot - 1);

quickSort(list, pivot + 1, end);

}

}

В основе быстрой сортировки лежит стратегия «разделяй и властвуй». Задача разделяется на более мелкие подзадачи. Подзадачи решаются отдельно, а потом решения объединяют. Точно так же, массив разделяется на подмассивы, которые сортируются и затем сливаются в один.