**Анализ сложности поразрядной сортировки**

Один из алгоритмов, которые не используют сравнение элементов массива. Сначала элементы сортируются по своему младшему(последнему) разряду, затем следующему(предпоследнему) и т.д. до старшего разряда, первого.  
До сортировки необходимо знать два параметра: k и d, где:  
k - количество разрядов в самом длинном ключе  
d - разрядность данных: количество возможных значений разряда ключа

|  |  |  |
| --- | --- | --- |
| **Лучшая** | **Средняя** | **Худшая** |
| - | n \* k/d | n \* k/d |

void SortRadix(int y) { // Поразрядная сортировка

vector <int> sup1 = main;

vector <vector<int>> sup2;

int c = 1, m = 10;

for (int i = 0; i < 10; i++)

sup2.push\_back(vector<int>());

for (int i = 0; i < 3; i++) {

for (int a = 0; a < main.size(); a++)

sup2[(sup1[a] % m) / c].push\_back(sup1[a]);

sup1.clear();

for (int a = 0; a < 10; a++)

for (int j = 0; j < sup2[a].size(); j++)

sup1.push\_back(sup2[a][j]);

for (int i = 0; i < sup2.size(); i++)

sup2[i].clear();

m \*= 10;

c \*= 10;

}

return sup1;

}







