Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №8 по курсу «Дискретный анализ»

Студент: А. А. Литвина
Преподаватель: А. А. Кухтичев
Группа: М8О-206Б
Дата:
Оценка:
Подпись:

Москва, 2019

Лабораторная работа №8

Задача: Разрабтать жадный алгоритм решения задачи, определяемой своим вариантом.
5. Оптимальная сортировка чисел
Дана последовательность длины N из целых чисел 1, 2, 3. Необходимо найти минимальное количество обменов элементов последовательности,в результате которых последовательность стала бы отсортированной.
Входные данные: число N на первой строке и N чисел на второй строке.
Выходные данные: минимальное количество обменов.

1 Описание

Жадный алгоритм — алгоритм, заключающийся в принятии локально оптимальных решений на каждом этапе, допуская, что конечное решение также окажется оптимальным.
Сложность жадного алгоритма для моей задачи - $O(n^2)$, а наивного алгоритма - $O(n!)$.

2 Исходный код

Считаем количество единиц и двоек в массиве и, исходя из этого, делим его на три части. В первой части идем слева направо, ищем сначала двойку, если нашли, тогда во второй и третьей части идем слева направо и ищем единицу, если нашли, меняем их местами. Если элемент оказался не двойкой, то ищем тройку, если нашли, тогда во второй и третьей части идем справа налево и ищем единицу, если нашли, меняем местами. Далее во второй части ищем тройку, если нашли, то в третьей части идем слева направо и ищем двойку, если нашли, меняем местами. При каждом обмене увеличиваем счетчик количества обменов. Это и есть наш результат.

```
#include <iostream>
#include <cstdio>
#include <vector>

using namespace std;

int main() {
  vector <int> v;
  int N;
  int a;
  int count1=0;
  int count2=0;
  int lim1;
  int lim2;
  int swap=0;
  cin >> N;
  for (int i=0; i<N; i++){
    cin >> a;
    if (a==1) count1++;
    else if(a==2) count2++;
    v.push_back(a);
  }

  lim1=count1;
  lim2=count1+count2;

  for (int i=0; i<lim1; i++) {
    if (v[i]==2) {
      for (int j=lim1; j<N; j++) {
        if (v[j]==1) {
          v[i]=1;
          v[j]=2;
          swap++;
          break;
        }
      }
```

```
    }
    else if (v[i]==3) {
      for (int j=N-1; j>=lim1; j--) {
        if (v[j]==1) {
          v[i]=1;
          v[j]=3;
          swap++;
          break;
        }
      }
    }
  }

  for (int i=lim1; i<lim2; i++) {
    if (v[i]==3) {
      for (int j=lim2; j<N; j++) {
        if (v[j]==2) {
          v[i]=2;
          v[j]=3;
          swap++;
          break;
        }
      }
    }
  }

  cout << swap << endl;
}
```

3 Консоль

```
anast@anast-Lenovo-B590:~$ ./a.out
3
3 2 1
1
anast@anast-Lenovo-B590:~$ ./a.out
7
3 2 3 1 2 1 2
3
anast@anast-Lenovo-B590:~$
```

4 Тест производительности

Тест производительности представляет из себя 10 цифр. Сравниваю с "наивным"алгоритмом.

```
anast@anast-Lenovo-B590:~$ ./a.out | grep "time"
Greedy time: 0,0001
Standart time: 3,6288
anast@anast-Lenovo-B590:~$
```

Как видно, жадный алгоритм работает намного быстрее "наивного особенно на больших данных.

5 Выводы

В данной лабораторной работе я узнала, что такое жадный алгоритм и как его использовать на практике. Это весьма интересный и эффективный метод решения задач, однако он имеет ряд недостатков. Во-первых, далеко не любую задачу можно решить с помощью жадного алгоритма, а во-вторых, на примере моей задачи, жадный алгоритм становится эффективным только на больших входных данных, в противном случае, если данных совсем мало, "наивный"алгоритм оказывается быстрее. Но, чем больше количество входных данных, тем быстрее растет эффективность данного метода.

Список литературы

Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн. *Алгоритмы: построение и анализ, 2-е издание.* — Издательский дом «Вильямс», 2007. Перевод с английского: И. В. Красиков, Н. А. Орехова, В. Н. Романов. — 1296 с. (ISBN 5-8459-0857-4 (рус.))

Жадный алгоритм — Википедия.
URL: `https://ru.wikipedia.org/wiki/Жадный_алгоритм`.