**Министерство образования и науки Российской Федерации**

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,   
МЕХАНИКИ И ОПТИКИ**

Факультет программной инженерии и компьютерной техники  
Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Aлгоритмы и структуры данных»

**ОТЧЁТ**

по лабораторной работе №10 (Week 10 Openedu)

Студенка Кузенкова Елизавета группы P3217

Преподаватель Муромцев Дмитрий Ильич

Санкт-Петербург

2019 г.

Содержание

[Задача 1 Префикс функция 3](#_Toc8423819)

[Исходный код к задаче 1 3](#_Toc8423820)

[Бенчмарк к задаче 1 4](#_Toc8423821)

[Задача 2. Z-функция 5](#_Toc8423822)

[Исходный код к задаче 2 6](#_Toc8423823)

[Бенчмарк к задаче 2 6](#_Toc8423824)

[Задача 3. Декомпозиция строки 8](#_Toc8423825)

[Исходный код к задаче 3 8](#_Toc8423826)

[Бенчмарк к задаче 3 10](#_Toc8423827)

# Задача 1 Префикс функция

|  |  |
| --- | --- |
| Имя входного файла: | input.txt |
| Имя выходного файла: | output.txt |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 256 мегабайт |

Постройте префикс-функцию для всех непустых префиксов заданной строки s.

#### Формат входного файла

Первая строка входного файла содержит . Строка состоит из букв латинского алфавита.

#### Формат выходного файла

Выведите значения префикс-функции для всех префиксов строки s длиной 1, 2, …, s в указанном порядке.

#### Примеры

|  |  |
| --- | --- |
| input.txt | output.txt |
| aaaAAA | 0 1 2 0 0 0 |
| abacaba | 0 0 1 0 1 2 3 |

## Исходный код к задаче 1

class Lab10\_1

{

public static void Main(string[] args)

{

var app = new Lab10\_1();

app.DoWork(args);

}

public static int[] PrefFunc(string s)

{

int i, j, n = s.Length;

int[] pi = new int[n];

for (i = 1; i < n; ++i)

{

j = pi[i - 1];

while (j > 0 && s[i] != s[j]) j = pi[j - 1];

if (s[i] == s[j]) ++j;

pi[i] = j;

}

return pi;

}

private void DoWork(string[] args)

{

using (StreamWriter sw = new StreamWriter("output.txt"))

{

string[] stdin = File.ReadAllLines("input.txt");

var numbers = PrefFunc(stdin[0]);

for (int i = 0; i < numbers.Length; i++)

{

sw.Write(numbers[i] + " ");

}

}

}

}

## Бенчмарк к задаче 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ теста** | **Результат** | **Время, с** | **Память** | **Размер входного файла** | **Размер выходного файла** |
| Max |  | 0.250 | 23322624 | 1000002 | 6888890 |
| 1 | OK | 0.031 | 9994240 | 8 | 12 |
| 2 | OK | 0.031 | 10141696 | 9 | 14 |
| 3 | OK | 0.031 | 10067968 | 3 | 2 |
| 4 | OK | 0.031 | 9977856 | 4 | 4 |
| 5 | OK | 0.031 | 10002432 | 4 | 4 |
| 6 | OK | 0.015 | 10043392 | 12 | 20 |
| 7 | OK | 0.015 | 9998336 | 12 | 20 |
| 8 | OK | 0.046 | 11792384 | 92672 | 185340 |
| 9 | OK | 0.046 | 11866112 | 99998 | 588846 |
| 10 | OK | 0.046 | 11886592 | 100002 | 561029 |
| 11 | OK | 0.062 | 12222464 | 176391 | 352778 |
| 12 | OK | 0.078 | 12382208 | 199994 | 1288779 |
| 13 | OK | 0.062 | 12390400 | 199992 | 1190917 |
| 14 | OK | 0.062 | 12333056 | 172864 | 345724 |
| 15 | OK | 0.093 | 13012992 | 300002 | 1988890 |
| 16 | OK | 0.078 | 13070336 | 300002 | 1716599 |
| 17 | OK | 0.062 | 12746752 | 249367 | 498730 |
| 18 | OK | 0.109 | 13615104 | 400002 | 2688855 |
| 19 | OK | 0.109 | 13590528 | 399998 | 2333024 |
| 20 | OK | 0.109 | 13910016 | 455342 | 910680 |
| 21 | OK | 0.109 | 14348288 | 499996 | 3388798 |
| 22 | OK | 0.125 | 14323712 | 499998 | 2875816 |
| 23 | OK | 0.109 | 20246528 | 505139 | 1010274 |
| 24 | OK | 0.156 | 20852736 | 600000 | 4088811 |
| 25 | OK | 0.156 | 20836352 | 600002 | 3977735 |
| 26 | OK | 0.125 | 20496384 | 539096 | 1078188 |
| 27 | OK | 0.187 | 21483520 | 699998 | 4788807 |
| 28 | OK | 0.156 | 21495808 | 700002 | 4566500 |
| 29 | OK | 0.109 | 14225408 | 492073 | 984142 |
| 30 | OK | 0.187 | 22093824 | 799997 | 5488765 |
| 31 | OK | 0.187 | 22048768 | 800002 | 3984418 |
| 32 | OK | 0.156 | 21848064 | 763540 | 1527076 |
| 33 | OK | 0.203 | 22700032 | 899994 | 6188744 |
| 34 | OK | 0.203 | 22720512 | 900002 | 5662193 |
| 35 | OK | 0.171 | 22306816 | 836144 | 1672284 |
| 36 | OK | 0.234 | 23285760 | 1000002 | 6888855 |
| 37 | OK | 0.218 | 23322624 | 1000002 | 6555315 |
| 38 | OK | 0.218 | 23289856 | 1000002 | 2000000 |
| 39 | OK | 0.250 | 23277568 | 1000002 | 6888890 |
| 40 | OK | 0.234 | 23240704 | 1000002 | 6888890 |
| 41 | OK | 0.218 | 23285760 | 1000002 | 6209773 |
| 42 | OK | 0.234 | 23236608 | 1000002 | 6888885 |
| 43 | OK | 0.234 | 23261184 | 1000002 | 6888870 |
| 44 | OK | 0.234 | 23257088 | 1000002 | 6388895 |
| 45 | OK | 0.234 | 23289856 | 1000002 | 6883895 |

# Задача 2. Z-функция

|  |  |
| --- | --- |
| Имя входного файла: | input.txt |
| Имя выходного файла: | output.txt |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 256 мегабайт |

Постройте Z-функцию для заданной строки s.

#### Формат входного файла

Первая строка входного файла содержит . Строка состоит из букв латинского алфавита.

#### Формат выходного файла

Выведите значения Z-функции для всех индексов 2, 3, .., s строки s в указанном порядке.

#### Примеры

|  |  |
| --- | --- |
| input.txt | output.txt |
| aaaAAA | 2 1 0 0 0 |
| abacaba | 0 1 0 3 0 1 |

## Исходный код к задаче 2

class Lab10\_2

{

public static void Main(string[] args)

{

var app = new Lab10\_2();

app.DoWork(args);

}

public static int[] ZFunc(string s)

{

int n = s.Length;

int[] z = new int[n];

for (int i = 1, l = 0, r = 0; i < n; ++i)

{

if (i <= r)

z[i] = Math.Min(r - i + 1, z[i - l]);

while (i + z[i] < n && s[z[i]] == s[i + z[i]])

++z[i];

if (i + z[i] - 1 > r)

{

l = i;

r = i + z[i] - 1;

}

}

return z;

}

private void DoWork(string[] args)

{

using (StreamWriter sw = new StreamWriter("output.txt"))

{

string[] stdin = File.ReadAllLines("input.txt");

var numbers = ZFunc(stdin[0]);

for (int i = 1; i < numbers.Length; i++)

{

sw.Write(numbers[i] + " ");

}

}

}

}

## Бенчмарк к задаче 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ теста** | **Результат** | **Время, с** | **Память** | **Размер входного файла** | **Размер выходного файла** |
| Max |  | 0.250 | 23314432 | 1000002 | 6888888 |
| 1 | OK | 0.031 | 10256384 | 8 | 10 |
| 2 | OK | 0.031 | 10235904 | 9 | 12 |
| 3 | OK | 0.031 | 10272768 | 4 | 2 |
| 4 | OK | 0.015 | 10203136 | 4 | 2 |
| 5 | OK | 0.031 | 10211328 | 5 | 4 |
| 6 | OK | 0.046 | 10199040 | 12 | 18 |
| 7 | OK | 0.046 | 10264576 | 12 | 18 |
| 8 | OK | 0.046 | 11980800 | 92672 | 185338 |
| 9 | OK | 0.046 | 12042240 | 99998 | 264801 |
| 10 | OK | 0.046 | 11972608 | 100002 | 272211 |
| 11 | OK | 0.046 | 12464128 | 176391 | 352776 |
| 12 | OK | 0.062 | 12591104 | 199994 | 474050 |
| 13 | OK | 0.062 | 12414976 | 199992 | 456479 |
| 14 | OK | 0.062 | 12201984 | 172864 | 345722 |
| 15 | OK | 0.093 | 13000704 | 300002 | 1988888 |
| 16 | OK | 0.109 | 13033472 | 300002 | 786113 |
| 17 | OK | 0.078 | 12693504 | 249367 | 498728 |
| 18 | OK | 0.093 | 13611008 | 400002 | 1036108 |
| 19 | OK | 0.093 | 13565952 | 399998 | 885168 |
| 20 | OK | 0.109 | 13918208 | 455342 | 910678 |
| 21 | OK | 0.109 | 14274560 | 499996 | 1217153 |
| 22 | OK | 0.125 | 14258176 | 499998 | 1267986 |
| 23 | OK | 0.125 | 20271104 | 505139 | 1010272 |
| 24 | OK | 0.140 | 20848640 | 600000 | 1406340 |
| 25 | OK | 0.140 | 20840448 | 600002 | 1477776 |
| 26 | OK | 0.125 | 20471808 | 539096 | 1078186 |
| 27 | OK | 0.156 | 21475328 | 699998 | 1682393 |
| 28 | OK | 0.156 | 21479424 | 700002 | 1558331 |
| 29 | OK | 0.109 | 14217216 | 492073 | 984140 |
| 30 | OK | 0.187 | 22097920 | 799997 | 1804662 |
| 31 | OK | 0.171 | 22118400 | 800002 | 2196116 |
| 32 | OK | 0.187 | 21868544 | 763540 | 1527074 |
| 33 | OK | 0.171 | 22740992 | 899994 | 2030971 |
| 34 | OK | 0.203 | 22659072 | 900002 | 2765564 |
| 35 | OK | 0.171 | 22331392 | 836144 | 1672282 |
| 36 | OK | 0.218 | 23248896 | 1000002 | 2611108 |
| 37 | OK | 0.218 | 23289856 | 1000002 | 2227775 |
| 38 | OK | 0.203 | 23248896 | 1000002 | 1999998 |
| 39 | OK | 0.250 | 23298048 | 1000002 | 6888888 |
| 40 | OK | 0.234 | 23310336 | 1000002 | 6888888 |
| 41 | OK | 0.234 | 23298048 | 1000002 | 2841971 |
| 42 | OK | 0.250 | 23277568 | 1000002 | 4444443 |
| 43 | OK | 0.203 | 23285760 | 1000002 | 2977776 |
| 44 | OK | 0.203 | 23314432 | 1000002 | 2000043 |
| 45 | OK | 0.203 | 23257088 | 1000002 | 2004885 |

# Задача 3. Декомпозиция строки

|  |  |
| --- | --- |
| Имя входного файла: | input.txt |
| Имя выходного файла: | output.txt |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 256 мегабайт |

Строка ABCABCDEDEDEF содержит подстроку ABC, повторяющуюся два раза подряд, и подстроку DE, повторяющуюся три раза подряд. Таким образом, ее можно записать как ABC\*2+DE\*3+F, что занимает меньше места, чем исходная запись той же строки.

Ваша задача - построить наиболее экономное представление данной строки s в виде, продемонстрированном выше, а именно, подобрать такие s1,a1 .. sk, ak где si - строки, аi - числа, чтобы s = s1 \* a1 + .. + sk \* ak. Под операцией умножения строки на целое положительное число подразумевается конкатенация одной или нескольких копий строки, число которых равно числовому множителю, то есть, ABC\*2=ABCABC. При этом требуется минимизировать общую длину итогового описания, в котором компоненты разделяются знаком +, а умножение строки на число записывается как умножаемая строка и множитель, разделенные знаком \*. Если же множитель равен единице, его, вместе со знаком \*, допускается не указывать.

#### Формат входного файла

Первая строка входного файла содержит . Строка состоит из букв латинского алфавита.

#### Формат выходного файла

Выведите оптимальное представление строки, данной во входном файле. Если оптимальных представлений несколько, выведите любое.

#### Примеры

|  |  |
| --- | --- |
| input.txt | output.txt |
| ABCABCDEDEDEF | ABC\*2+DE\*3+F |
| Hello | Hello |

## Исходный код к задаче 3

public class Lab10\_3

{

private static string[] \_input;

private static int \_currentLineIndex;

private static string ReadLine()

{

return \_input[\_currentLineIndex++];

}

public static void Main(string[] args)

{

var app = new Lab10\_3();

app.DoWork(args);

}

private void DoWork(string[] args)

{

\_input = File.ReadAllLines("input.txt");

string str = ReadLine();

int[] dim = new int[str.Length + 1];

int[] substrStart = new int[str.Length + 1];

int[] length = new int[str.Length + 1];

for (int i = 0; i < str.Length + 1; i++)

{

dim[i] = int.MaxValue;

}

dim[0] = 0;

for (int i = 0; i < str.Length; i++)

{

int[] p = new int[str.Length + 1];

p[1] = 0;

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

{

dim[i + 1] = dim[i] + 1;

substrStart[i + 1] = i;

length[i + 1] = 1;

}

int k = 0;

for (int j = 2; i + j - 1 < str.Length; j++)

{

while (str[i + j - 1] != str[i + k] && k > 0)

{

k = p[k];

}

if (str[i + j - 1] == str[i + k])

{

k++;

}

p[j] = k;

if (j % (j - p[j]) == 0)

{

if (dim[i + j] > dim[i] + (j - p[j]))

{

dim[i + j] = dim[i] + (j - p[j]);

substrStart[i + j] = i;

length[i + j] = j - p[j];

}

}

}

}

StringBuilder sb = new StringBuilder();

string[] substring = new string[str.Length];

int[] substringsCount = new int[str.Length];

int align = 0;

for (int i = str.Length; i > 0;)

{

substring[align] = str.Substring(substrStart[i], length[i]);

substringsCount[align] = (i - substrStart[i]) / length[i];

align++;

i = substrStart[i];

}

int maxAlign = align - 1;

bool isPreviousAppendable = true;

for (align--; align >= 0; align--)

{

bool optimized = (substring[align].Length > 2 || substringsCount[align] != 2) && (substring[align].Length != 1 || substringsCount[align] != 3);

if (optimized)

{

bool isOldAppendable = isPreviousAppendable;

isPreviousAppendable = substringsCount[align] <= 1;

if ((align != maxAlign) && (!isOldAppendable || !isPreviousAppendable))

{

sb.Append("+");

}

sb.Append(substring[align]);

if (!isPreviousAppendable)

{

sb.Append("\*");

sb.Append(substringsCount[align]);

}

}

else

{

string o = substring[align] + substring[align];

if (substringsCount[align] == 3)

{

o += substring[align];

}

if (!isPreviousAppendable)

{

sb.Append("+");

}

sb.Append(o);

isPreviousAppendable = true;

}

}

File.WriteAllText("output.txt", sb.ToString());

}

}

## Бенчмарк к задаче 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ теста** | **Результат** | **Время, с** | **Память** | **Размер входного файла** | **Размер выходного файла** |
| Max |  | 0.265 | 11976704 | 5002 | 5000 |
| 1 | OK | 0.031 | 10096640 | 15 | 12 |
| 2 | OK | 0.031 | 10153984 | 7 | 5 |
| 3 | OK | 0.015 | 10100736 | 3 | 1 |
| 4 | OK | 0.046 | 10059776 | 4 | 2 |
| 5 | OK | 0.031 | 10076160 | 5 | 3 |
| 6 | OK | 0.031 | 10088448 | 6 | 3 |
| 7 | OK | 0.015 | 10100736 | 7 | 3 |
| 8 | OK | 0.031 | 10076160 | 8 | 3 |
| 9 | OK | 0.031 | 10121216 | 9 | 3 |
| 10 | OK | 0.031 | 10113024 | 10 | 3 |
| 11 | OK | 0.031 | 10104832 | 11 | 3 |
| 12 | OK | 0.031 | 10100736 | 12 | 4 |
| 13 | OK | 0.015 | 10084352 | 13 | 4 |
| 14 | OK | 0.046 | 10051584 | 14 | 4 |
| 15 | OK | 0.015 | 10051584 | 4 | 2 |
| 16 | OK | 0.031 | 10108928 | 6 | 4 |
| 17 | OK | 0.015 | 10104832 | 8 | 4 |
| 18 | OK | 0.031 | 10108928 | 10 | 4 |
| 19 | OK | 0.015 | 10137600 | 12 | 4 |
| 20 | OK | 0.031 | 10113024 | 14 | 4 |
| 21 | OK | 0.015 | 10133504 | 16 | 4 |
| 22 | OK | 0.031 | 10067968 | 18 | 4 |
| 23 | OK | 0.031 | 10178560 | 20 | 4 |
| 24 | OK | 0.031 | 10104832 | 22 | 5 |
| 25 | OK | 0.031 | 10104832 | 24 | 5 |
| 26 | OK | 0.015 | 10108928 | 26 | 5 |
| 27 | OK | 0.031 | 10903552 | 450 | 448 |
| 28 | OK | 0.031 | 11292672 | 498 | 19 |
| 29 | OK | 0.031 | 11288576 | 498 | 56 |
| 30 | OK | 0.031 | 11276288 | 817 | 815 |
| 31 | OK | 0.031 | 11251712 | 1001 | 13 |
| 32 | OK | 0.031 | 11243520 | 1001 | 18 |
| 33 | OK | 0.046 | 11259904 | 1452 | 1450 |
| 34 | OK | 0.046 | 11309056 | 1502 | 6 |
| 35 | OK | 0.109 | 11292672 | 1502 | 73 |
| 36 | OK | 0.062 | 11268096 | 1828 | 1826 |
| 37 | OK | 0.062 | 11292672 | 1993 | 15 |
| 38 | OK | 0.078 | 11288576 | 1991 | 269 |
| 39 | OK | 0.031 | 11227136 | 1338 | 1336 |
| 40 | OK | 0.093 | 11386880 | 2499 | 15 |
| 41 | OK | 0.062 | 11333632 | 2486 | 188 |
| 42 | OK | 0.078 | 11272192 | 2402 | 2400 |
| 43 | OK | 0.109 | 11304960 | 3002 | 7 |
| 44 | OK | 0.093 | 11366400 | 2985 | 66 |
| 45 | OK | 0.062 | 11259904 | 2374 | 2372 |
| 46 | OK | 0.140 | 11337728 | 3502 | 14 |
| 47 | OK | 0.125 | 11317248 | 3500 | 10 |
| 48 | OK | 0.078 | 11317248 | 2227 | 2225 |
| 49 | OK | 0.171 | 11599872 | 4002 | 12 |
| 50 | OK | 0.171 | 11673600 | 4002 | 35 |
| 51 | OK | 0.109 | 11309056 | 2921 | 2919 |
| 52 | OK | 0.203 | 11403264 | 4502 | 22 |
| 53 | OK | 0.203 | 11431936 | 4501 | 86 |
| 54 | OK | 0.218 | 11849728 | 4765 | 4763 |
| 55 | OK | 0.234 | 11915264 | 5000 | 8 |
| 56 | OK | 0.250 | 11923456 | 4988 | 109 |
| 57 | OK | 0.250 | 11894784 | 5002 | 5000 |
| 58 | OK | 0.250 | 11505664 | 5002 | 6 |
| 59 | OK | 0.265 | 11976704 | 5002 | 31 |
| 60 | OK | 0.250 | 11964416 | 5002 | 82 |
| 61 | OK | 0.234 | 11915264 | 5002 | 7 |
| 62 | OK | 0.234 | 11927552 | 5002 | 10 |
| 63 | OK | 0.250 | 11497472 | 5002 | 503 |
| 64 | OK | 0.234 | 11501568 | 5002 | 54 |