

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего
образования

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИТМО

Факультет систем управления и робототехники

Лабораторная работа № 5
"Задачи 1067, 1521, 1628"

по дисциплине Алгоритмы и структуры данных

Выполнила: студентка гр. **R3238**
поток **2.1**

Нечаева А. А.

Преподаватель: *Тропченко Андрей Александрович*

Санкт-Петербург, 2024

1 Цель

Разработать и реализовать алгоритмы для решения задач 1067, 1521 и 1628.

2 Задача 1067

1067. Структура папок

Ограничение времени: 2.0 секунды
Ограничение памяти: 64 МБ

Хакер Билл случайно потерял всю информацию с жесткого диска своего компьютера, и у него нет резервных копий его содержимого. Но он сожалеет не о потере самих файлов, а о потере очень понятной и удобной структуры папок, которую он создавал и сохранял в течение многих лет работы.

К счастью, у Билла есть несколько копий списков папок с его жесткого диска. С помощью этих списков он смог восстановить полные пути к некоторым папкам (например, «WINNT\SYSTEM32\CERTSRV\CERTCO~1\X86»). Он поместил их все в файл, записав каждый найденный путь в отдельную строку.

Напишите программу, которая восстановит структуру папок Билла и выведет ее в виде отформатированного дерева.

Исходные данные

Первая строка содержит целое число N – количество различных путей к папкам ($1 \leq N \leq 500$). Далее следуют N строк с путями к папкам. Каждый путь занимает одну строку и не содержит пробелов, в том числе, начальных и конечных. Длина каждого пути не превышает 80 символов. Каждый путь встречается в списке один раз и состоит из нескольких имен папок, разделенных обратной косой чертой («\»).

Имя каждой папки состоит из 1-8 заглавных букв, цифр или специальных символов из следующего списка: восклицательный знак, решетка, знак доллара, знак процента, амперсанд, апостроф, открывающаяся и закрывающаяся скобки, знак дефиса, собаки, ширкумфлекс, подчеркивание, гравис, открывающаяся и закрывающаяся фигурная скобка и тильда («!#\$%&'()*~+-_{}|»).

Результат

Выведите отформатированное дерево папок. Каждое имя папки должно быть выведено в отдельной строке, перед ним должно стоять несколько пробелов, указывающих на глубину этой папки в иерархии. Подпапки должны быть перечислены в лексикографическом порядке непосредственно после их родительской папки; перед их именем должно стоять на один пробел больше, чем перед именем их родительской папки. Папки верхнего уровня выводятся без пробелов и также должны быть перечислены в лексикографическом порядке.

Пример

исходные данные	результат
7 WINNT\SYSTEM32\CONFIG GAMES WINNT\DRIVERS HOME WIN\SOFT GAMES\DRIVERS WINNT\SYSTEM32\CERTSRV\CERTCO~1\X86	GAMES DRIVERS HOME WIN SOFT WINNT DRIVERS SYSTEM32 CERTSRV CERTCO~1 X86 CONFIG

Источник задачи: 2000-2001 ACM Northeastern European Regional Programming Contest
Метки: структуры данных (скрыть метки для нерешенных задач)

Рис. 1. Условие задачи 1067.

2.1 Краткое описание алгоритма

1. Входные данные: первая строка содержит целое число N — количество путей к папкам ($1 \leq N \leq 500$). Далее следуют N строк с путями к папкам. Каждый путь занимает одну строку и не содержит пробелов, в том числе, начальных и конечных. Длина каждого пути не превышает 80 символов. Каждый путь встречается в списке один раз и состоит из нескольких имен папок, разделенных обратной косой чертой (" \backslash ").

Имя каждой папки состоит из 1-8 заглавных букв, цифр или специальных символов из следующего списка: восклицательный знак, решетка, знак доллара, знак процента, амперсанд, апостроф, открывающаяся и закрывающаяся скобки, знак дефиса, собаки, циркумфлекс, подчеркивание, гравис, открывающаяся и закрывающаяся фигурная скобка и тильда.

2. Прямая реализация описанного в задании алгоритма действий с помощью специальной структуры, содержащей имя и указатели на дочерние каталоги;

3. После реализован рекурсивный обход полученной структуры директорий в глубину.

4. И с помощью отдельного метода выведен результат.

5. Выходные данные: вывести отформатированное дерево папок. Каждое имя папки должно быть выведено в отдельной строке, перед ним должно стоять несколько пробелов, указывающих на глубину этой папки в иерархии. Подпапки должны быть перечислены в лексикографическом порядке непосредственно после их родительской папки; перед их именем должно стоять на один пробел больше, чем перед именем их родительской папки. Папки верхнего уровня выводятся без пробелов и также должны быть перечислены в лексикографическом порядке.

2.2 Листинг

Листинг 1. Исходный код для 1067

```
1 #include <iostream>
2 #include <map>
3 #include <utility>
4
5
6 // structure for smart store disk items and their relates
7 struct disk {
8     std::string item;
9     std::map<std::string, disk *> child;
```

```
10
11 // clang-tidy recommendation for single-argument
    constructions
12 explicit disk(std::string name) {
13     // clang-tidy recommendation
14     this -> item = std::move(name);
15     this -> child = {};
16 }
17
18 disk() {
19     this -> item = "";
20     this -> child = {};
21 }
22 };
23
24
25 // method for printing disk structure in recursive mode
26 void disk_printer(disk* cur, int depth) {
27     for (int i = 0; i < depth - 1; ++i) {
28         std::cout << " ";
29     }
30     if (!cur -> item.empty()) {
31         std::cout << cur -> item << std::endl;
32     }
33     ++depth;
34
35     for (auto &child_dir : cur -> child) {
36         disk_printer(child_dir.second, depth);
37     }
38 }
39
40
41 int main() {
42     int n;
43     std::cin >> n;
44
45     disk* rt = new disk();
46
47     for (int i = 0; i < n; ++i) {
48         std::string cur_path;
49         std::cin >> cur_path;
50         disk* cur_d = rt;
51         std::string d;
```

```

52
53     for (int j = 0; j <= cur_path.size(); ++j) {
54
55         if (cur_path[j] == '\\\\' || cur_path[j] == '\\0') {
56             auto d_disk = cur_d -> child.find(d);
57
58             if (d_disk == cur_d -> child.end()) {
59                 disk* new_disk = new disk(d);
60                 cur_d -> child[d] = new_disk;
61                 cur_d = cur_d -> child.find(d) -> second;
62
63             } else {
64                 cur_d = d_disk -> second;
65             }
66
67             d = "";
68
69         } else {
70             d += cur_path[j];
71         }
72     }
73 }
74 disk_printer(rt, 0);
75
76 return 0;
77 }

```

2.3 Результат

Timus Online Judge			
Online Judge	Задачи	Авторы	Соревнования
О системе	Архив задач	Регистрация	Текущее соревнование
Часто задаваемые вопросы	Отправить на проверку	Исправить данные	Расписание
Новости сайта	Состояние проверки	Ребрен автор	Промежуточные соревнования
Форум	Руководство	<input type="text" value="Имя автора"/> <input type="button" value="Поиск"/>	Правила
Ссылки			

Результаты проверки решений

ID	Дата	Автор	Задача	Язык	Результат проверки	№ теста	Время работы	Выделено памяти
10612759	22:06:28 8 апр 2024	mshDzaron	1067. Структура лапок	Clang++ 17 x64	Accepted		0.062	3 864 КБ

Рис. 2. Результат отправки задачи 1067.

3 Задача 1521

1521. Военные учения 2

Ограничение времени: 1.0 секунды

Ограничение памяти: 64 МБ

Вступление

В ходе недавних военных учений (более подробно эта история рассказана в задаче «[Военные учения](#)») министр обороны Советской Федерации товарищ Иванов имел возможность лично убедиться в блестящей боевой готовности солдат вверенной ему Советской Армии. Но одна вещь всё же продолжала беспокоить выдающегося военачальника. Прославленный генерал понимал, что была продемонстрирована лишь физическая подготовка солдат. Теперь настало время организовать очередные учения и проверить интеллектуальные способности личного состава.

Генерал Шульман, вновь назначенный ответственным за проведение учений, пожертвовал все выделенные деньги бедным и с чистой совестью лёг спать. Во сне генералу явился учебник по тактике и изложил схему, руководствуясь которой можно провести учения совершенно бесплатно.

Задача

В соответствии с этой схемой учения делятся на N раундов, в течение которых N солдат, последовательно пронумерованных от 1 до N , маршируют друг за другом по кругу, т.е. первый следует за вторым, второй за третьим, ..., $(N-1)$ -й за N -м, а N -й за первым. В каждом раунде очередной солдат выбывает из круга и идёт чистить унитазы, а оставшиеся продолжают маршировать. В очередном раунде выбывает солдат, марширующий на K позиций впереди выбывшего на предыдущем раунде. В первом раунде выбывает солдат с номером K .

Разумеется, г-н Шульман не питал никаких надежд на то, что солдаты в состоянии сами определить очерёдность выбывания из круга. «Эти неучи даже траву не могут ровно покрасить», – фыркнул он и отправился за помощью к прапорщику Шкурко.

Исходные данные

Единственная строка содержит целые числа N ($1 \leq N \leq 100000$) и K ($1 \leq K \leq N$).

Результат

Вывести через пробел номера солдат в порядке их выбывания из круга.

Пример

исходные данные	результат
5 3	3 1 5 2 4

Автор задачи: Илья Гребнов, Никита Рыбак, Дмитрий Ковалёв

Источник задачи: Timus Top Coders: Third Challenge

Рис. 3. Условие задачи 1521.

3.1 Краткое описание алгоритма

1. **Входные данные:** единственная строка содержит целые числа N ($1 \leq N \leq 100000$) и K ($1 \leq K \leq N$).

2. Просто запишем все номера солдат в массив, в том порядке, в котором

они идут по кругу. Далее будем удалять k -й элемент и заново рассчитывать номер элемента, который нужно удалить следующим.

3. Выходные данные: вывести через пробел номера солдат в порядке их выбывания из круга.

3.2 Листинг

Листинг 2. Исходный код для 1521

```
1 #include <iostream>
2 #include <vector>
3
4
5 int main() {
6     int n, k;
7     std::cin >> n >> k;
8     std::vector<int> circle_order;
9     circle_order.reserve(n);    // clang-tidy recommendation
10
11     for (int i = 0; i < n; ++i) {
12         circle_order.push_back(i);
13     }
14
15     int go_away = -k;    // position to go out of the circle
16
17     for (int i = -n; i > 0; --i) {
18         std::cout << circle_order[go_away] + 1 << " ";
19         circle_order.erase(circle_order.begin() + go_away);
20         go_away = (go_away + k) % i;
21     }
22
23     std::cout << circle_order[0] + 1 << " ";
24     return 0;
25 }
```

3.3 Результат

Timus Online Judge			
Online Judge	Задачи	Авторы	Соревнования
О системе Часто задаваемые вопросы Новости сайта Форум Ссылки	Архив задач Отправить на проверку Состояние проверки Руководство	Регистрация Исправить данные Рейтинг авторов <input type="text" value="Имя автора"/> <input type="button" value="Поиск"/>	Текущее соревнование Расписание Прошедшие соревнования Правила

Результаты проверки решений

ID	Дата	Автор	Задача	Язык	Результат проверки	№ теста	Время работы	Выделено памяти
10612787	22:59:56 8 апр 2024	miniDragon	1521. Военные учения 2	Clang++ 17 x64	Accepted		0.875	756 KB

Рис. 4. Результат отправки задачи 1521.

4 Задача 1628

1628. Белые полосы

Ограничение времени: 1.0 секунды

Ограничение памяти: 64 МБ

У каждого неудачника в жизни бывают не только чёрные, но и белые полосы. Марспанин Вась-Вась отмечает в календаре, представляющем собой таблицу $m \times n$, те дни, когда ему ужасно не повезло. Если Вась-Васю не повезло в j -й день i -й недели, то он закрашивает ячейку таблицы (i, j) в чёрный цвет. Все незакрашенные ячейки в таблице имеют белый цвет.

Будем называть отрезками жизни прямоугольнички размером $1 \times l$ либо $l \times 1$. Белыми полосами Вась-Вась считает все максимальные по включению белые отрезки таблицы. А сможете ли Вы определить, сколько всего белых полос было в жизни Вась-Вася?

Исходные данные

Первая строка содержит целые числа m, n, k — размеры календаря и количество неудачных дней в жизни Вась-Вася ($1 \leq m, n \leq 30000$; $0 \leq k \leq 60000$). В следующих k строках перечислены неудачные дни в виде пар (x_i, y_i) , где x_i — номер недели, к которой относится неудачный день, а y_i — номер дня в этой неделе ($1 \leq x_i \leq m$; $1 \leq y_i \leq n$). Описание каждого неудачного дня встречается только один раз.

Результат

Выведите число белых полос в жизни Вась-Вася.

Примеры

исходные данные	результат
3 5 4 1 1 1 5 2 2 3 3	8
5 1 2 2 1 3 1	2

Автор задачи: Александр Ипатов

Источник задачи: XIII Открытый командный чемпионат УрГУ по программированию

Рис. 5. Условие задачи 1628.

4.1 Краткое описание алгоритма

1. **Входные данные:** первая строка содержит целые числа m, n, k — размеры календаря и количество неудачных дней в жизни Вась-Вася ($1 \leq m, n \leq 30000$; $0 \leq k \leq 60000$). В следующих k строках календаря перечислены неудачные дни в виде пар (x_i, y_i) , где x_i — номер недели, к которой относится неудачный день, а y_i — номер дня в этой неделе ($1 \leq x_i \leq m$; $1 \leq y_i \leq n$). Описание каждого неудачного дня встречается только один раз.

2. Допишем черные дни в начале / в конце каждой строчки и столбца.
3. Отсортируем все точки, начав в левом верхнем углу и закончив в правом нижнем: сначала по строкам, потом по столбцам. Все полосы, длина которых больше 1, добавим к результату. Отдельно сохраним все полосы, длина которых равна единице. Так же отсортируем еще раз все точки, но в этот раз сначала столбцы, а затем строки. Так мы получим число полос по вертикали и по горизонтали.
4. Теперь проверим полосы 1×1 : если такая полоса содержится в векторе ровно 2 раза, то тогда получается, что квадрат 1×1 является максимальным по горизонтали и вертикали, значит, добавляем его в результат.
5. Выходные данные: вывести число белых полос в жизни Вася-Вася.

4.2 Листинг

Листинг 3. Исходный код для 1628

```
1 #include <iostream>
2 #include <bits/stdc++.h>
3
4
5 // to compare first items
6 bool c_1(std::pair<int, int> a, std::pair<int, int> b) {
7     return (a.first != b.first) ? a.first < b.first : a.second
8         < b.second;
9 }
10
11 //to compare second items
12 bool c_2(std::pair<int, int> a, std::pair<int, int> b) {
13     return (a.second != b.second) ? a.second < b.second : a.
14         first < b.first;
15 }
16
17 int main() {
18     std::vector<std::pair<int, int>> day;
19     std::vector<std::pair<int, int>> s_day;
20     int m, n, k;
21     int ans = 0;
22
23     std::cin >> m >> n >> k;
24
25     for (int i = 0; i < k; ++i) {
26         std::pair<int, int> cur;
```

```
25         std::cin >> cur.first >> cur.second;
26         cur.first--;
27         cur.second--;
28         day.push_back(cur);
29     }
30
31     for (int i = 0; i < m; ++i) {
32         std::pair<int, int> cur = {i, -1};
33         std::pair<int, int> c_cur = {i, n};
34         day.push_back(c_cur);
35         day.push_back(cur);
36     }
37
38     for (int i = 0; i < n; ++i) {
39         std::pair<int, int> cur = {-1, i};
40         std::pair<int, int> c_cur = {m, i};
41         day.push_back(c_cur);
42         day.push_back(cur);
43     }
44     std::sort(day.begin(), day.end(), c_1);
45
46     for (int i = 0; i < day.size() - 1; ++i) {
47         if (day[i].first == day[i + 1].first) {
48             if (day[i + 1].second - day[i].second > 2) {
49                 ++ans;
50             } else if (day[i + 1].second - day[i].second == 2) {
51                 std::pair<int, int> cur(day[i].first, day[i].
52                     second + 1);
53                 s_day.push_back(cur);
54             }
55         }
56
57         std::sort(day.begin(), day.end(), c_2);
58
59         for (int i = 0; i < day.size(); ++i) {
60             if (day[i].second == day[i + 1].second) {
61                 if (day[i + 1].first - day[i].first > 2) {
62                     ++ans;
63                 } else if (day[i + 1].first - day[i].first == 2) {
64                     std::pair<int, int> cur(day[i].first + 1, day[
65                         i].second);
```

```

65         s_day.push_back(cur);
66     }
67 }
68 }
69
70 std::sort(s_day.begin(), s_day.end(), c_1);
71
72 if (s_day.size() > 1) {
73     for (int i = 0; i < s_day.size() - 1; ++i) {
74         if (s_day[i] == s_day[i + 1]) {
75             ++ans;
76             ++i;
77         }
78     }
79 }
80
81 std::cout << ans;
82 return 0;
83
84 }

```

4.3 Результат

Timus Online Judge			
Online Judge	Задачи	Авторы	Соревнования
О системе	Архив задач	Регистрация	Текущее соревнование
Часть задаваемых вопросов	Отправить на проверку	Послать решение	Расписание
Новости сайта	Состояние проверки	Рейтинг авторов	Прошедшие соревнования
Форум	Руководство	<input type="text" value="Имя автора"/> <input type="button" value="Поиск"/>	Правила
Ссылки			

Результаты проверки решений

ID	Дата	Автор	Задача	Язык	Результат проверки	№ теста	Время работы	Выделено памяти
10612007	06:31:40 9 апр 2024	misiDragon	1628. Белые полосы	Clang++ 17 x64	Accepted		0.25	2 756 КБ

Рис. 6. Результат отправки задачи 1628.

5 Вывод по работе

В ходе выполнения данной лабораторной работы были реализованы алгоритмы для решения задач 1067, 1521 и 1628. Использованные структуры данных: динамические массивы и структуры.