

Университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа №3
по «Алгоритмам и структурам данных»

Выполнил:

Студент группы Р3210

Машина Екатерина

Преподаватели:

Косяков М.С.

Санкт-Петербург

2020

Задача №1067 «Disk Tree»

https://github.com/mashinakatherina/Algorithms_and_Data_Structures/blob/master/Part3/t1067.cpp

Пояснение к примененному алгоритму:

Для решения задачи нам необходимо представить данные в следующем формате:

- Массив в каждой ячейке которого хранится map
- Map хранит информацию о дочерних каталогах в виде пары ключ - значение, где ключ - имя дочерней директории для текущего каталога, а значение - ссылка на описание дочернего каталога (такую же структуру).

Визуализация примера:

```
WINNT\SYSTEM32\CONFIG
GAMES
WINNT\DRIVERS
HOME
WIN\SOFT
GAMES\DRIVERS
WINNT\SYSTEM32\CERTSRV\CERTCO~1\X86
```

\	WINNT	GAMES	HOME	WIN
WINNT	DRIVERS	SYSTEM32		
SYSTEM32	CERTSRV	CONFIG		
DRIVERS				
GAMES	DRIVERS			
DRIVERS				
HOME				
WIN	SOFT			
CONFIG				
SOFT				
CERTSRV	CERTCO~1			
CERTCO~1	X86			
X86				

Структура:

```
GAMES
DRIVERS
HOME
WIN
SOFT
WINNT
DRIVERS
SYSTEM32
CERTSRV
CERTCO~1
X86
CONFIG
```

Задача №1494 «Монобильярд»

https://github.com/mashinakatherina/Algorithms_and_Data_Structures/blob/master/Part3/t1494.cpp

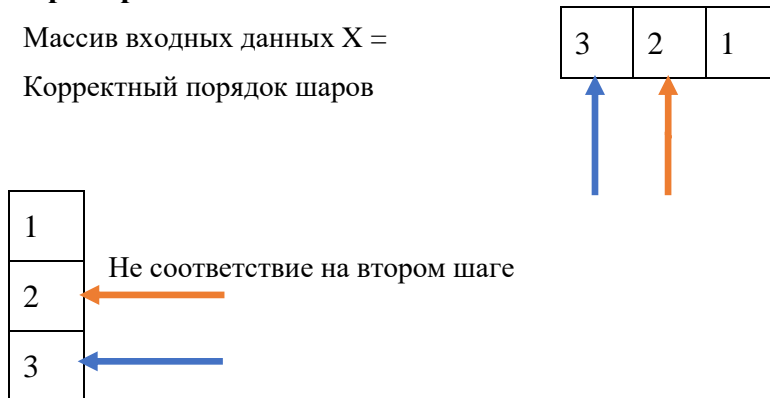
Пояснение к примененному алгоритму:

Попробуем положить шары в стек так же, как их забивал бы Чичиков (таким образом на вершине стека окажется последний шар).

Далее начнем снимать со стека шары и проверяем порядок в нашем массиве (проверка на то, что текущий шар строго больше всех предыдущих). Если порядок совпадает, то шары забиты в верном порядке, потому что именно в этом порядке шары попадают в лунку. Если нет, то соответственно Чичиков жульничает.

Пример:

Массив входных данных $X =$
Корректный порядок шаров



Задача №1521 «Военные учения 2»

https://github.com/mashinakatherina/Algorithms_and_Data_Structures/blob/master/Part3/t1521.cpp

Пояснение к примененному алгоритму:

Для решения задачи используем дерево отрезков (так как связный список совсем не проходит по времени).

С его помощью будем хранить сколько солдат осталось на конкретном отрезке. Поиск и удаление в дереве происходит за $O(\log n)$, т.к. необходимо дойти до корня.

Также задачу можно решить при помощи дерева Фенвика, которое также позволяет искать каждый k -ый элемент за $O(\log n)$.

Задача №1628 «Белые полосы»

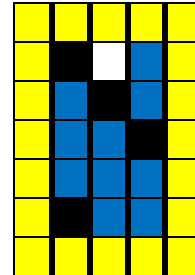
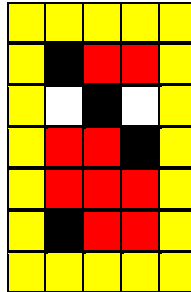
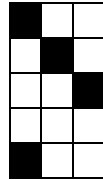
https://github.com/mashinakatherina/Algorithms_and_Data_Structures/blob/master/Part3/t1628.cpp

Пояснение к примененному алгоритму:

- Добавляем контур для исходной матрицы (чтобы при «разворачивании матрицы строки и столбцы были разделены»)
- Записываем все строки в один ряд (по строкам и столбцам отдельно (см. пример ниже))
- В разложении матрицы по строкам находим все разрывы, длина которых больше 1
- Все единичные разрывы складываем в Set.
- Аналогичные действия проводим с разложением по столбцам.
- При нахождении единичных разрывов учитываем их только в том случае, если они уже присутствуют в Set.

Пример:

3 5 4
1 1
1 5
2 2
3 3



Сортировка по строкам:



Сортировка по столбцам:



Задача №1650 «Миллиардеры»

https://github.com/mashinakatherina/Algorithms_and_Data_Structures/blob/master/Part3/t1650.cpp

Пояснение к примененному алгоритму:

Для решения задачи нам понадобится 2 структуры:

Отсортированное по возрастанию множество городов, в котором храним актуальное на данный день состояние городов. При каждом перелете миллиардера изменяем общую сумму денег в городе и обновляем наше множество.

Во второй структуре данных будем хранить данные об имени миллиардера и городе в котором он сейчас находится. Это удобнее всего сделать при помощи map, где ключом будет имя миллиардера, а значением структура хранящая количество денег, которые находятся в городе и сколько дней они там находятся.

Все миллиардеры и города хранятся в map, для того, чтобы изменения какой-либо информации совершалось за логарифм.