ЛОШ ФМЛ информатика, день 3

24 августа 2022

1 Мощности компьютеров

2 Временная асимптотика алгоритмов

3 std::vector

```
#include<vector>
int n = 10; // размер вектора
vector<int> A(n , 1) // 1 - изначально заполняемый элемент
A.push_back(2) // добавить в конец 2
```

4 Битовые операции

1. побитовый **OR**
2. побитовый **AND**
3. побитовый **XOR**
4. побитовый **NOT**
5. сдвиг влево и сдвиг вправо

5 Задача на кратность

Как проверить делится ли число a на 2^n не используя % (остаток от деления)?

Ответ: (a | (1 « n)) == (1 « n) или ((a » n) « n) == a

6 Представление множеств

Все числа в памяти представлены в двоичной системе исчесления.
Идея: пусть элементы множества это позиции (в 0-нумерации) единичных битов.Тогда каждое число задает свое кникальное множество
Пример: $1242 = 10011011010_2$ задает множество $\{1, 3, 4, 6, 7, 10\}$ (позиции едичных битов)

7 Подпоследовательность и подотрезок массива

Подотрезок массива это массив который может получен путем удаление элементов из **начала** и или **конца** исходного массива.

Подпоследовательность массива это массив который может получен путем удаление элементов из **любого** места исходного массива.

8 Задача о количестве объектов

Дан массив A длиной n, нужно найти количество его подпоследовательностей и количество подотрезков.

Ответ: 2^n и $\frac{n(n+1)}{2}$

9 Задачи о количестве объектов с заданными храктеристиками

Дан массив A , $A_i \leq 10^9$ длиной $n \leq 10^3$, нужно найти нужно найти количество его количество подотрезков с четной суммой.

Решите предыдущую задачу при $n \leq 10^5$.

Дан массив A, $A_i \leq 10^9$ длиной $n \leq 20$, нужно найти количество его количество подпоследовательностей с четной суммой.

(hard)Решите предыдущую задачу при $n \leq 10^5$ и найдите остакток от деления ответа на $10^9 + 7$ или же решите задачу по модулю $10^9 + 7$.
Примечание: $10^9 + 7$ простое число.

10 Модульная арифметика

$(a + b)\%MOD = (a\%MOD + b\%MOD)\%MOD$

$(a * b)\%MOD = (a\%MOD * (b\%MOD))\%MOD$

$(a - b)\%MOD = (a\%MOD - b\%MOD)\%MOD$

11 Быстрое возведение в степень по модулю

$a^n = (a^{\frac{n}{2}})^2$ если n четное.
$a^n = aa^{(n-1)}$ если n нечетное.
$a^n = 1$ если n = 0.
все опепрации кончно же по модулю

```
int pw(int a, int b, int MOD){
    if(!b) return 1;
    if(b % 2) return (1ll * a * pw(a ,  b - 1 , MOD)) % MOD;
    int x = pw(a , b / 2, MOD);
    return (1ll * x * x) % MOD;
}
```

12 Малая теорема Ферма (МТФ)

Если p — простое число и a — целое число, не делящееся на p, то $a^{p-1} - 1$ делится на p.
Более формально: $a^{p-1} - 1 \equiv 0 \mod p$.
Добавим 1 к левой и правой части: $a^{p-1} \equiv 1 \mod p$

13 Деление по простому модулю

$b/a \equiv b * a^{-1} \mod MOD$ теперь нам нужно найти a^{-1}, то есть такое число которое при умножении на a дает 1 вернемся к МТФ: $a^{p-1} \equiv 1 \mod p$ и представим $a^{p-1} = aa^{p-2}$ тогда: $aa^{p-2} \equiv 1 \mod p$ следовательно $a^{p-2} \equiv a^{-1} \mod p$ и получаем равентсво :

$$b/a \equiv ba^{MOD-2} \mod MOD$$

14 Комбинаторные объекты по модулю

$\binom{n}{k} \equiv \frac{n!}{k!(n-k)!} \equiv n!(k!(n-k)!)^{MOD-2} \mod MOD$

15 Основная теорема арифметики (ОТА)

Любое натуральное число больше единицы может быть разложено в виде простых множителей и это разложение единственно (если не учитывать порядок множителей).

16 Решето Эратосфена

Оценка временной сложности:
$\mathcal{O}(\sum\limits_{i=1}^{n} \lfloor \frac{n}{i} \rfloor) = \mathcal{O}(n \sum\limits_{i=1}^{n} \frac{1}{i}) = \mathcal{O}(n \int\limits_{1}^{n} \frac{1}{x}\, dx) = \mathcal{O}(n \ln n) = \mathcal{O}(n \log n)$

17 Алгоритм Евклида

```
int gcd(int a, int b) {
    if (b == 0)
        return a;
    return gcd(b, a % b);
}
```

или встроенная функия

```
#include<algorithm>
__gcd(a , b);
```

18 Задача про НОД

Дан массив A , $A_i \leq 10^9$ длиной $n \leq 10^5$ найти НОД (наибольший общий делитель) всех его элементов.
Оцените временную сложность вашего алгоритма.

Ответ: $\mathcal{O}(n + \log(\max A_i))$ а не $\mathcal{O}(n \log(\max A_i))$!!!

19 (hard) Задача про НОД

Дан массив A , $A_i \leq 10^5$ длиной $n \leq 10^5$ найти максимальнай НОД по всем возможным парам элементов, пары из одинаковых индексов не считать.

Подсказка №1: ограничение на A_i меньше чем обычно.
Подсказка №2: подумайте об $\mathcal{O}(C \log C + n)$ или $\mathcal{O}(C + n\sqrt{C})$ или об $\mathcal{O}(C + n\sqrt[3]{C})^*$, где $C = \max A_i$.

20 Задача о поиске подотрезка с максимальной суммой

Пусть дан массив целых чисел A , $|A_i| \leq 10^9$ длиной $n \leq 10^2$ найти отрезок с максимальной суммой.
Решите эту же задачу при $n \leq 5 * 10^3$.
Решите эту же задачу при $n \leq 10^6$.

21 std::set и std::map

```
#include<set>
set<int> A;
A.insert(2); // добавить 2
A.erase(2); // удалть 2
cout << A.count(2); // проверить на существование
```

22 (Hard) Для тех кому скучно №1.

Дан массив натуральных чисел A , $A_i \leq 2 * 10^7$ длиной $n \leq 5 * 10^4$ разрешается прибавлять или вычетать из любого элемента по 1 , но количество операций не должно привосходить n какой максимальный НОД всех элементов вы сможете получить?
Ограничение: 3 секунды и 2048 мБ.

Подсказка №1: Сколько элементов изменяются не более чем на 1?
Подсказка №2: Если мы знаем какое число находится в оптимальном массиве, как узнать ответ?
Подсказка №3: Что там по метеоритам?

23 (HARD) Для тех кому скучно №2.

Даны три натуральных числа a , b и p такие что $a, b < p$ и p - **простое**. Разрешается делать над a дествия трех видов:

1. заменить a на $(a+1) \mod p$
2. заменить a на $(a-1) \mod p$
3. заменить a на $a^{p-2} \mod p$

Нужно за 300 операций (можно меньше) превратить a в b. Требется найти последовательность действий. Если ответов несколько выбирете любой.
Для каких ограничений на p вы умеете решать данную задачу с ограничением по времени 3 секунды и по памяти 2048 мБ?

Подсказка №1: Вы же написали динамическое программирование?
Подсказка №2: Забудьте о ДП, подумайте о днях рождения и о BFS.