Алгоритмы 12

Максим Пасько

6 мая 2018 г.

Задание 1

Будем выводить последовательности в порядке возрастания. Тогда первой будет последовательность $x[1] = ... = x[k] = 1$, а последней - $x[1] = ... = x[k] = n$. В массиве $prev$ будем хранить предыдущий массив, изначально равный $prev[1] = ... = prev[k] = n$.

Следующая последовательность будет находится следующим образом:
1) Находим максимальный s такой, что $x[s] < n$.
2) Все элеменеты, позиция которых больше s, приравниваем к 1, а сам $x[s]$ увеличиваем на 1.

В итоге, алгоритм будет таким:
1) $X = 11...11$
2) $prev = nn...nn$
3) **while**$(X \neq prev)$ {
4) **print**(next(X)) (тут next(X) - функция, строющая следующую после X последовательность)
5) $prev = X$ }

Асимптотика такого алгоритма будет $O(n^k k)$.

Задание 2

Пусть у нас изначально $\omega = u_1 u_2 ... u_k$. Найдём минимальную по сумме пару u_i, u_{i+1}. После этого шага $\omega = u_1 u_2 ... v_i ... u_k$, где $v_i = u_i u_{i+1}$. Теперь рассматриваем v_i как u_i, то есть, с v_i тоже можно составлять такие пары. Таким образом, дойдём до того момента, когда $\omega = v_k v_l$. Мы каждый шаг спаривали минимальные по сумме пары, таким образом, если идти с конца, то есть первым разрезом разделить ω на v_k и v_l, а далее идти по следующим парам, то сумма всех операций будет минимальна, ведь мы выбрали пары минимальной суммы.

Асимптотика такого алгоритма будет $O(k^2)$, ибо поиск минимальной пары стоит $O(k)$ времени, и мы на каждом из $k - 1$ шагов ищем эту минимальную пару.

Задание 3

Заметим, что если количество карт каждого значения чётное (то есть, если количество карт значения i чётное $\forall i \in [1, n]$), то игрок, которой должен сделать ход на такую раскладку, проигрывает, ибо каждый раз он будет уменьшать количество карт на нечётное число, в конце-концов оставив своему сопернику 1 карту. Тогда правильной игрой будет процесс, при котором игрок пытается поставить на этот проигрышный расклад своего соперника. Тогда на самом первом ходу Алиса проигрывает в том случае, если количество карт каждого значения чётное. Иначе же она просто Возьмёт карту, удовлетворяющую таким условиям:
(1) Количество карт такого номинала - нечётное число;
(2) Номинал этой карты - это максимальное число из тех, что удовлетворяют условию (1).

Таким действием Алиса оставит ту самую проигрышную комбинацию Бобу.

Тогда узнать победителя мы можем таким образом:
1) Создадим массив $Count$ на n элементов, изначально заполненный нулями;
2) Пройдёмся по массиву карт, если номинал карты i, то $++Count[i]$;
3) Пройдём ещё раз по массиву $Count$ и если найдём i такое, что $Count[i]$ нечётное, тогда Алиса побеждает. Если же все $Count[i]$ чётные, то Алиса проиграла.

Асимптотика $O(n)$.

Задание 4

Если какое-то число S не представляется в виде сумма некоторых чисел массива, и к тому же является минимальным таким, то в этом случае все числа от 1 до $S-1$ представляются как сумма некоторых чисел массива. Тогда появляется инвариант, что сумма некоторых элементов массива $a[1]...a[k]$ может принимать все значения от 1 до некоторого значения N. При этом, если $N+1 < a[k+1]$, то $N+1$ и есть ответ, т.к. его представление как сумма элементов $a[1], ..., a[k]$ невозможно, а одно лишь $a[k+1]$ больше него. А если $N+1 \geq a[k+1]$, то сумма некоторых членов массива $a[1], ..., a[k+1]$ принимает все значения от 1 до $N + a[k+1]$. Итоговый алгоритм будет такой:
1) $N = 0$
2) $\mathbf{for}(k = 0; k < n; ++k)\{$

if$(a[k+1] > N+1)$
print$(N+1)$ and break
else $N = N + a[k+1]$ }
print$(N+1)$

Асимптотика такого алгоритма, очевидно, будет $O(n)$.

Задание 5

Пронумеруем вершины от 1 до n по часовой стрелке. Пусть $A(k,l)$ (k $>$ l) - это многоугольник, не содержащий ребра (1, n), получаемый разрезом исходного по хорде (k,l). $a(k,l)$ пусть будет означать стоймость разрезания $A(k,l)$ по диагоналям. При этом, если $l = k+1$ или $l = k+2$, то $a(k,l) = 0$, т.к. получаются "двуугольник"(одно ребро) и треугольник соответственно. Тогда получаем рекурентную формулу на минимальную величину $a(k,l)$:

$$min(|(k,i)| + |(i,l)| + a(k,i) + a(i,l))$$

по всем $i \in [k+1, l-1]$. Причём, т.к. мы считаем длину хорд, то $|k,i| = 0$, если $i - k = 1$. Тогда ответом будет минимальное $a(k,l)$ по количеству вершин n.

У нас будет $O(n^2)$ приминений рекуррентной формулы, которая требует выбора минимума из не более n элементов.