Университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа №1

по «Алгоритмам и структурам данных»

Базовые задачи / **Timus**

Выполнил:

Студент группы P32121

Гиниятуллин Арслан Рафаилович

Преподаватели:

Косяков М.С.

Тараканов Д.С.

Санкт-Петербург

2023

Задачи с Timus’a, handle = **vtalgo23_giniiatullin_335089**

Задача 1005: Куча камней

Решение за O(2^n * n) **bitmasks**:

```
void solve() {
    int n;
    cin >> n;
    vector<int> weights(n);
    int sum = 0;
    for (auto &i: weights) {
        cin >> i;
        sum += i;
    }
    int mn = sum;
    for (size_t mask = 0; mask < (1 << n); ++mask) {
        int sum1 = 0, sum2 = 0;
        for (size_t i = 0; i < n; ++i) {
            if ((mask >> i) & 1 | 0) {
                sum1 += weights[i];
            } else {
                sum2 += weights[i];
            }
        }
        mn = min(mn, abs(sum1 - sum2));
    }
    cout << mn << '\n';

}
```

Описание решения:

Переберем все случаи разделения массива на два непересекающихся множества и найдем минимальную разность между суммами элементов в них. Обыкновенная рекурсия, я реализовал, использовав битмаски.

Асимптотика: O(2^n * n), перебор вариантов за 2^n и n на поиск суммы для каждого.

Решение за O(n * SUM(0, n) / 2) **knapsack с ML☹:**

```
void solve() {
    int n;
    cin >> n;
    vector<int> weights(n);
    int sum = 0;
    for (auto &i: weights) {
        cin >> i;
        sum += i;
    }
```

```
    int dp[n + 1][(sum + 1) / 2 + 1];

    for (int c = 0; c <= sum; ++c) {
        dp[0][c] = 0;
    }

    for (int i = 1; i <= n; ++i) {
        for (int c = 0; c < weights[i - 1]; ++c) {
            dp[i][c] = dp[i - 1][c];
        }
        for (int c = weights[i - 1]; c <= (sum + 1) / 2; ++c) {
            dp[i][c] = max(dp[i - 1][c], dp[i - 1][c - weights[i - 1]] +
weights[i - 1]);
        }
    }
    cout << abs(dp[n][(sum + 1) / 2] - (sum - dp[n][(sum + 1) / 2])) << '\n';
}
```

Описание решения:

Решим задачу о рюкзаке, в котором ценность груза = весу. Тогда dp[i][w] – максимальная сумма из первых i весов с суммарным весом меньшим w. База очевидна, переход будет – максимум из предыдущих и, учитывая текущий вес. Найдем ответ для dp[n][(sum + 1) / 2] и выразим минимальную разность между множествами.

Асимптотика: O(n * SUM(0, n) / 2)

Не прошло по памяти, не догадался, как отсекать

Задача 1401: Игроки

```
ll result_table[512][512];
ll num = 0;

void color(ll x, ll y, ll i, ll j, ll n) {
    ++num;
    n = n / 2;
    if ((x - j) >= n && (y - i) >= n) {
        result_table[i + n][j + n - 1] = num;
        result_table[i + n - 1][j + n] = num;
        result_table[i + n - 1][j + n - 1] = num;
        if (n > 1) {
            color(j + n - 1, i + n - 1, i, j, n);
            color(j + n, i + n - 1, i, j + n, n);
            color(x, y, i + n, j + n, n);
            color(j + n - 1, i + n, i + n, j, n);
```

```
        }
    } else if ((x - j) < n && (y - i) < n) {
        result_table[i + n - 1][j + n] = num;
        result_table[i + n][j + n] = num;
        result_table[i + n][j + n - 1] = num;
        if (n > 1) {
            color(x, y, i, j, n);
            color(j + n, i + n - 1, i, j + n, n);
            color(j + n, i + n, i + n, j + n, n);
            color(j + n - 1, i + n, i + n, j, n);
        }
    } else if ((x - j) >= n && (y - i) < n) {
        result_table[i + n - 1][j + n - 1] = num;
        result_table[i + n][j + n] = num;
        result_table[i + n][j + n - 1] = num;
        if (n > 1) {
            color(j + n - 1, i + n - 1, i, j, n);
            color(x, y, i, j + n, n);
            color(j + n, i + n, i + n, j + n, n);
            color(j + n - 1, i + n, i + n, j, n);
        }
    } else if ((x - j) < n && (y - i) >= n) {
        result_table[i + n - 1][j + n - 1] = num;
        result_table[i + n - 1][j + n] = num;
        result_table[i + n][j + n] = num;
        if (n > 1) {
            color(j + n - 1, i + n - 1, i, j, n);
            color(j + n, i + n - 1, i, j + n, n);
            color(j + n, i + n, i + n, j + n, n);
            color(x, y, i + n, j, n);
        }
    }
    re;
}

ll binpow(ll a, ll n) {
    ll res = 1;
    while (n) {
        if (n & 1)
            res *= a;
        a *= a;
        n >>= 1;
    }
    return res;
}

void solve() {
    ll n, x, y;
    cin >> n >> x >> y;
    n = 1 << n;
    if ((ll) (binpow(n, 2) - 1ll) % 3 != 0) {
        cout << "-1\n";
        re;
    }

    color(x - 1, y - 1, 0, 0, n);

    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j) {
```

```
            cout << result_table[i][j] << ' ';
        }
        cout << '\n';
    }

}
```

Описание решения:

Решим задачку рекурсией. Будем каждый раз уменьшать сторону в два раза, проверять соответствующие условия для (x, y), чтобы покрасить в квадратики в нужные цвета, далее запускаемся из 3 наших точек и начальной (x, y) и продолжаем, пока сторона не будет единичной длины

Асимптотика: O(2^(n/2))