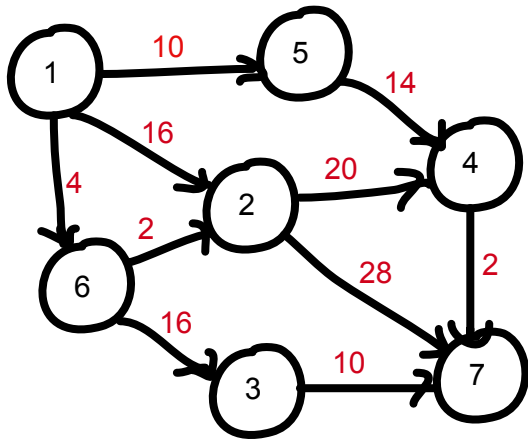


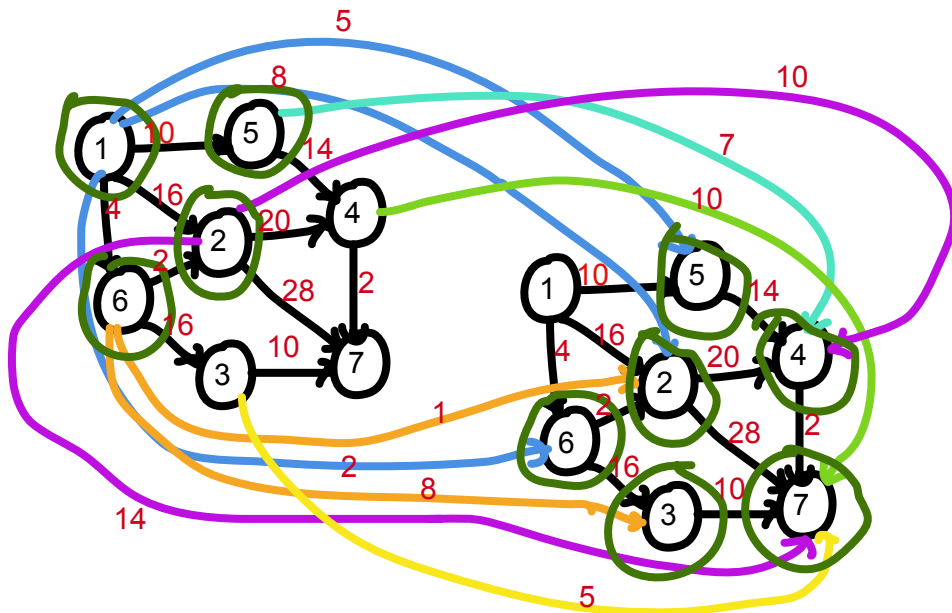
<https://cses.fi/problemset/task/1195>



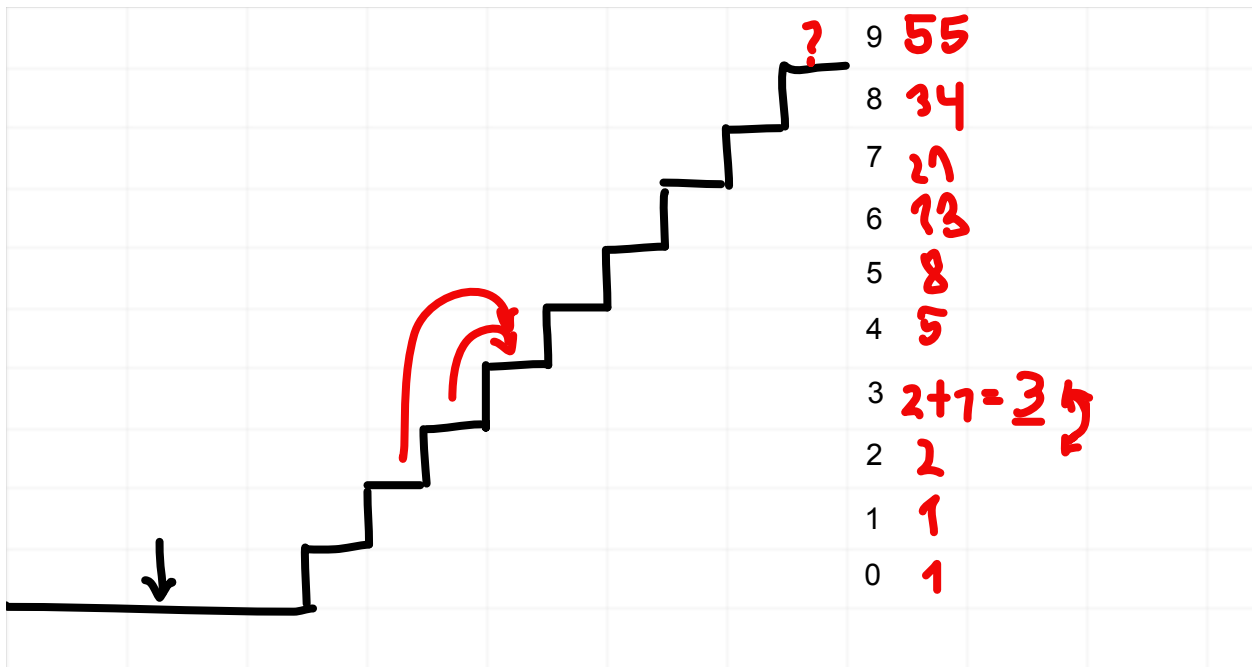
(X, discountUsed)

Колко е най-краткият път от (1, false) до (n, true)?

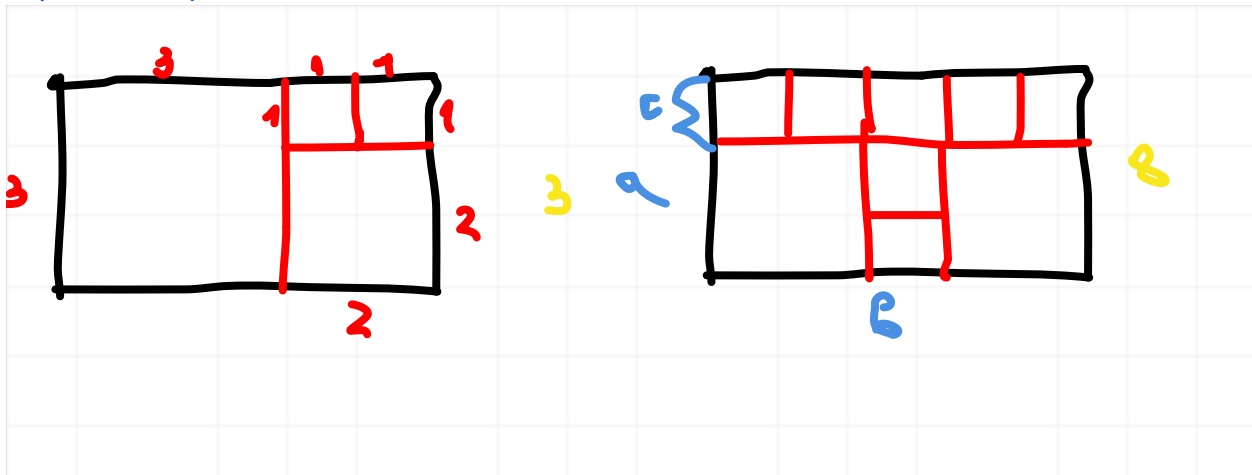
distance	1	2	3	4	5	6	7
unused	0*	6*	20	24	10*	4*	34
used	infinity	4*	12*	16*	5*	2*	18*



<https://leetcode.com/problems/climbing-stairs/>



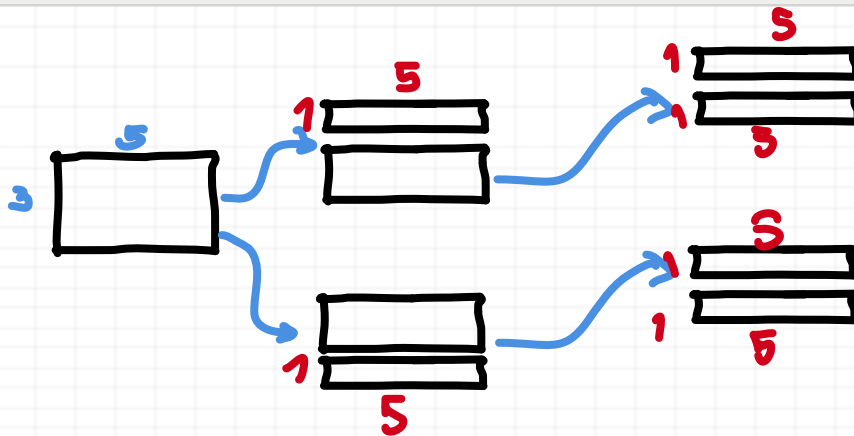
<https://cses.fi/problemset/task/1744>



```

int minCuts(int a, int b) {
    if (a == b) {
        return 0;
    }
    if (calculated[a][b] != -1) { // ако вече сме го смятали, направо
    връщаме запазена стойност
        return calculated[a][b];
    }
    int ans = infinity;
    for (int c = 1; c < a; c++) {
        // двата правоъгълника, които остават са c x b и (a-c) x b
        int cuts = 1 + minCuts(c, b) + minCuts(a-c, b);
        if (cuts < ans) {
            ans = cuts;
        }
    }
    for (int c = 1; c < b; c++) {
        // двата правоъгълника, които остават са a x c и a x (b-c)
        int cuts = 1 + minCuts(a, c) + minCuts(a, b-c);
        if (cuts < ans) {
            ans = cuts;
        }
    }
    calculated[a][b] = ans;
    return ans;
}

```



<https://cses.fi/problemset/task/1637>

495 → 491

495 → 490 → 486

495 → 486

```
vector<int> getDigits(int n) {
    vector<int> digits;
    int tmp = n;
    while (tmp > 0) {
        digits.append(tmp % 10);
        tmp = tmp / 10;
    }
    return digits;
}

int minSteps(int n) {
    if (n == 0) {
        return 0;
    }
    if (calculated[n] != -1) {
        return calculated[n];
    }
    vector<int> digits = getDigits(n);

    int ans = infinity
    for (digit in digits) {
        int steps = 1 + minSteps(n - digit);
        if (steps < ans) {
            ans = steps;
        }
    }
    calculated[n] = ans;
    return ans;
}
```

Задачи, които ще разгледаме следващия път

<https://cses.fi/problemset/task/1145>

<https://leetcode.com/problems/palindromic-substrings/>

<https://cses.fi/problemset/task/1639>