



Дивизион D. День 1 (Разбор)
Динамическое программирование

Лектор: Дмитрий Руденко

Задача А

Реализовать алгоритм из лекции

Задача G

- 1) "Засасывать" больше одного шарика не имеет смысла
- 2) Давайте найдем самую большую возрастающую подпоследовательность длины len
- 3) Остальные $(n - len)$ шариков которые в нее не входят переставим пылесосом на свои места за $(n - len)$ операций

Задача F

- 1) Давайте отсортируем гангстеров по времени прихода
- 2) Уже известным алгоритмом найдем НВП гангстеров
- 3) Осталось ввести понятие "возрастания" гангстеров
- 4) Гангстер А больше гангстера Б, если их разница во времени не меньше разницы в полноте
- 5) В массиве dp храним не длину подпоследовательности, а сумму богатств всех гангстеров подпоследовательности
- 6) Стоит учитывать, что последовательность из 1 гангстера может существовать, только если время прибытия гангстера не меньше его полноты

Задача В

```
vector<pair<int, int>> ans;
int size = 0;
for (int i = a.size(), j = b.size(); i > 0 || j > 0; ) {
    if (path[i][j] == pair{ i - 1, j - 1 }) {
        ans[size] = { i, j };
        size++;
    }
    pair<int, int> next = path[i][j];
    i = next.first;
    j = next.second;
}

reverse(ans, ans + size);
cout << size << endl;
for (int i = 0; i < size; i++) {
    cout << ans[i].first << " ";
}
cout << endl;
for (int i = 0; i < size; i++) {
    cout << ans[i].second << " ";
}
```

Задача С

- 1) Не забываем, что строки могут быть пустыми (в C++ `cin` не работает - используем `getline(cin, str)`)
- 2) Делаем оптимизацию: `ab**cd?g*cdfd**** = ab*cd?g*cdfd*`
- 3) `dp[i][j]` - ответ для префиксов `a[0...i - 1]` и `b[0...j - 1]` (true или false)
- 4) Изначально везде стоит false, и только `dp[0][0] = true`
- 5) Если `a[0] = *`, тогда заполняем `dp[1][j]` значение true по всем `j`
- 6) `a[i - 1] = ? => dp[i][j] = dp[i - 1][j - 1]`
- 7) `a[i - 1] = буква => если a[i - 1] == b[j - 1]`, тогда аналогично шагу 6, иначе `dp[i][j] = false`
- 8) `a[i - 1] = *`, `dp[i][j] = dp[i - 1][j] || dp[i][j - 1]`

Задача D

- 1) Если $i = 0, j = 0 \Rightarrow dp[i][j] = 0$
- 2) Если $j = 0, i > 0 \Rightarrow dp[i][j] = i$
- 3) Если $i = 0, j > 0 \Rightarrow dp[i][j] = j$
- 4) Если $a[i] == b[j], dp[i][j] = dp[i - 1][j - 1]$
- 5) Иначе $dp[i][j] = \min(dp[i, j - 1], dp[i - 1][j], dp[i - 1][j - 1]) + 1$

Задача E

1) Давайте жадно наберем 2 пилообразные последовательности

< > < > < ... > <

> < > < > ... > <

2) Ответ: $n - \max(\text{len1}, \text{len2})$

1 < 2 3 > 1 < 2 (4)

2 2 3 > 1 < 2 (3)

$\text{dp1}[i] \rightarrow \text{НПП} > a[j] < a[i]$

$\text{dp2}[i] \rightarrow \text{НПП} < a[j] > a[i]$

$i = [0..n)$

$\text{dp1}[i] = 1$

$\text{dp2}[i] = 1$

$j = [0..i) \rightarrow a[i] > a[j] \rightarrow \text{dp1}[i] = \max(\text{dp1}[i], \text{dp1}[j] + 1)$

$a[i] < a[j] \rightarrow \text{dp2}[i] = \max(\text{dp2}[i], \text{dp2}[j] + 1)$

$\text{len} = \max(\text{dp1}, \text{dp2})$

$\text{answer} = n - \text{len}$