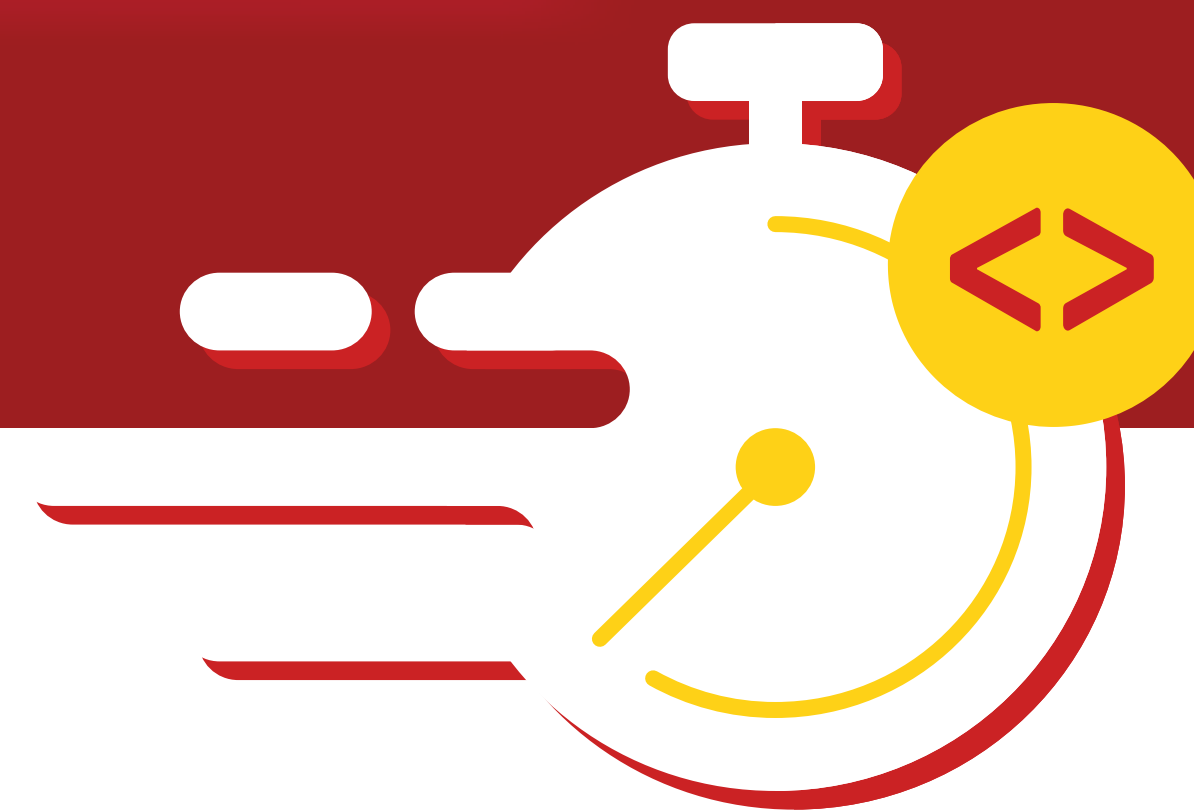
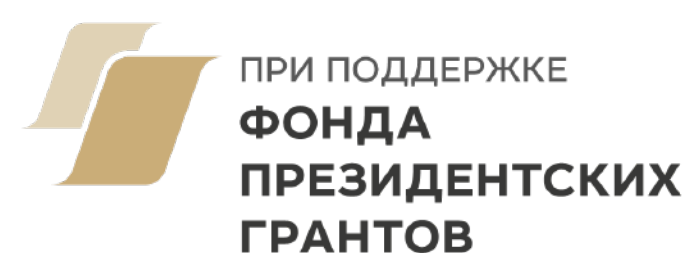


Восстановление ответа запоминанием предков

Урок 3.3



Задача восстановления ответа_

- При вычислении динамики запоминать клетку

Запоминание предков_

1	4	9	15
0	4	6	15
5	-6	-5	-9
-1	0	-2	-7

Восстановление ответа_

1	4	9	15
0	4	6	15
5	-6	-5	-9
-1	0	-2	-7

Реализация_

```
1  /*
2  ...
3  Объявление массивов a, dp
4  Чтение массива a
5  */
6
7  vector <vector <int> > parent_x(n + 1, vector <int> (m + 1, -1));
8  vector <vector <int> > parent_y(n + 1, vector <int> (m + 1, -1));
9
10 for (int i = 1; i <= n; ++i) {
11     for (int j = 1; j <= m; ++j) {
12         dp[i][j] = min(min(dp[i][j - 1], dp[i - 1][j]),
13                        dp[i - 1][j - 1]) + a[i][j];
14
15         if (dp[i][j] == dp[i - 1][j] + a[i][j]) {
16             parent_x[i][j] = i - 1;
17             parent_y[i][j] = j;
18         } else if (dp[i][j] == dp[i][j - 1] + a[i][j]) {
19             parent_x[i][j] = i;
20             parent_y[i][j] = j - 1;
21         } else {
22             parent_x[i][j] = i - 1;
23             parent_y[i][j] = j - 1;
24         }
25     }
26 }
```

Восстановление ответа_

```
1  int x = n, y = m;
2  vector <pair <int, int> > ans;
3
4  while (x > 0 && y > 0) {
5      ans.push_back({x, y});
6      int old_x = x, old_y = y;
7      x = parent_x[old_x][old_y];
8      y = parent_y[old_x][old_y];
9  }
10
11 for (int i = ans.size() - 1; i >= 0; --i) {
12     cout << ans[i].first << " " << ans[i].second << endl;
13 }
```

Итог

+

-

Просмотр
динамики


Чуть медленнее
восстановление

$O(1)$ дополнительной
памяти

Предки

Быстрое
восстановление

Тратим больше
памяти



Следующее занятие — задача о рюкзаке
