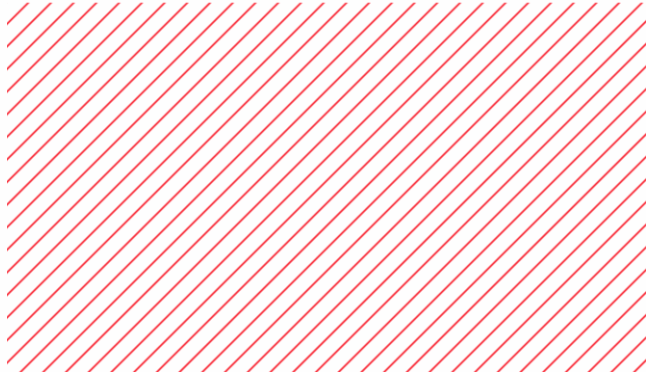
академия больших данных

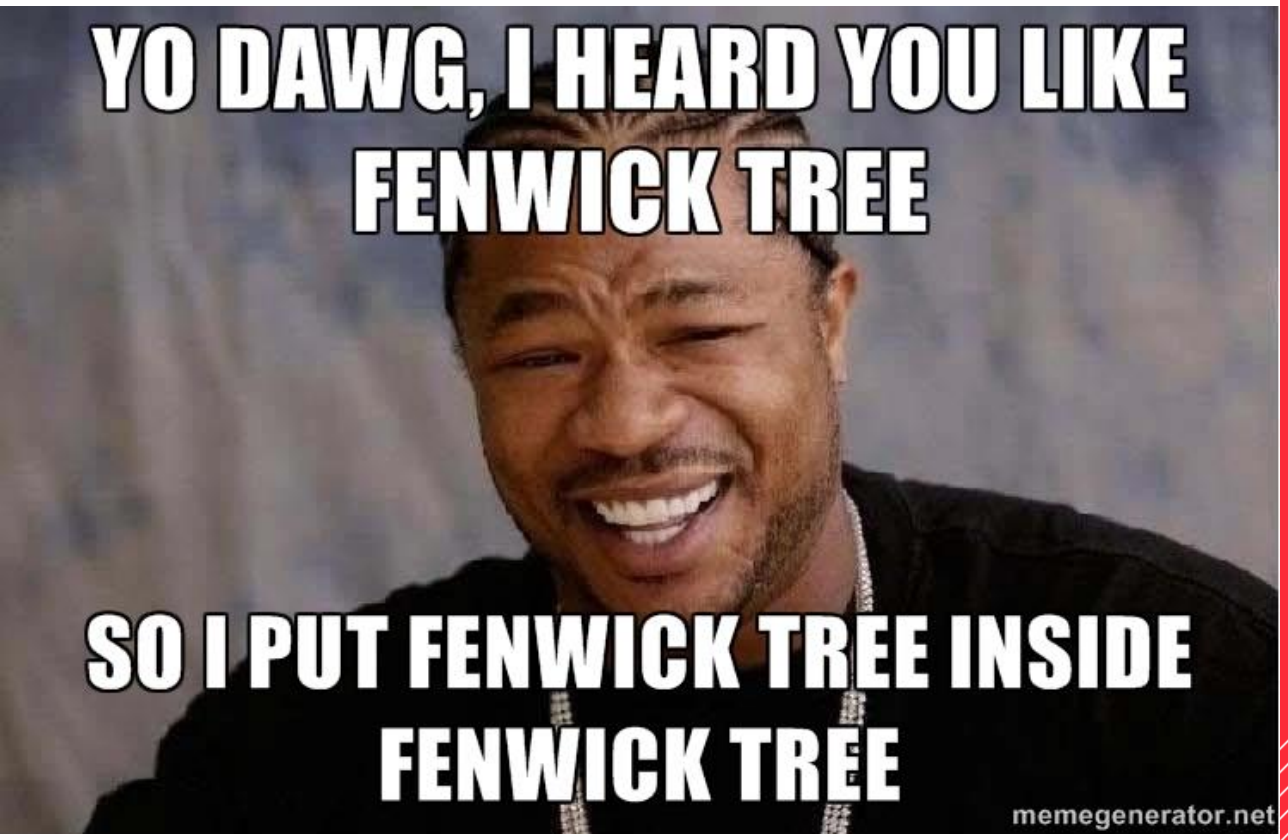
mail.ru group

Запросы на отрезках
Продолжение

Шовкопляс Григорий

Алгоритмы и структуры данных Advanced





Двумерные (многомерные) запросы

Многомерные запросы

- Есть **матрица** чисел, к ней можно последовательно применять запросы:
  - Сумма на подпрямоугольнике: $\sum_{i=L_1}^{R_1} \sum_{j=L_2}^{R_2} a[i]\,[j]$
  - Минимум/максимум на подпрямоугольнике
  - Изменить элемент: $\mathrm{a[i][j]} = x$
  - И другие…
- Куб (3d)
- Гиперкуб?

2d Префиксные суммы

Префиксные суммы

- Сможем за *O(1)* отвечать на запрос *RSQ(l1, r1, l2, r2)*
- Пусть дан массив *a[i][j]*
- Посчитаем массив *sum[i][j]* = $\sum_{i\prime=0}^{i}\sum_{j\prime=0}^{j} a[i'][j']$
  - Как считать не «в лоб»?
  - $sum[i][j] = sum[i-1][j] + sum[i][j-1] - sum[i-1][j-1] + a[i][j]$
- Чему равно *RSQ(l1, r1, l2, r2)*?
  - $sum[l2][r2] - sum[l1-1][r2] - sum[l2][r1-1] + sum[l1-1][r1-1]$

Многомерное дерево отрезков

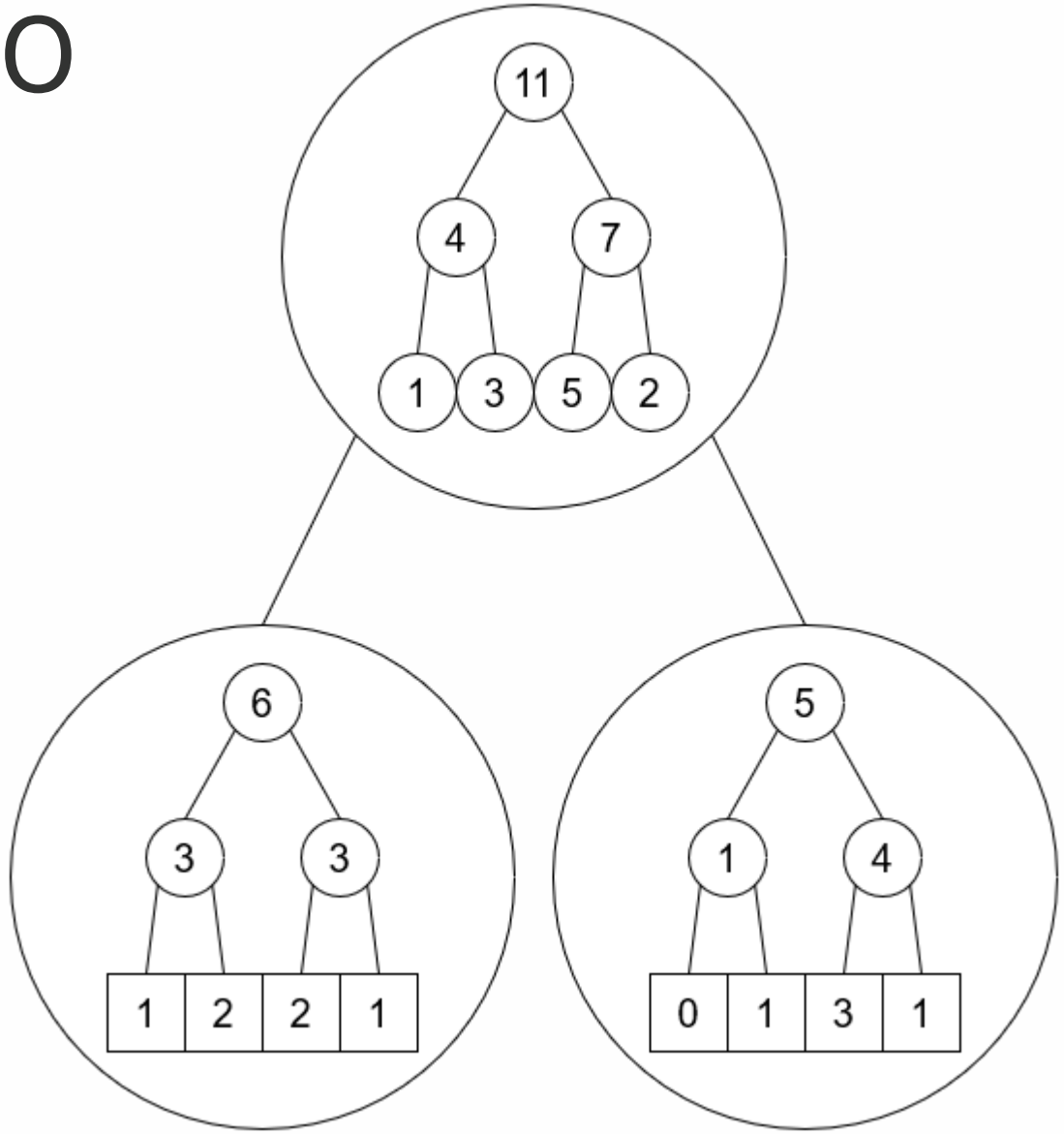
Многомерное ДО

- Вершина – дерево отрезков меньшей размерности
- Нижний уровень одномерное дерево отрезков для «массива»

- Работает для любой ассоциативной операции

Многомерное ДО

1	2	2	1
0	1	3	1



Многомерное ДО

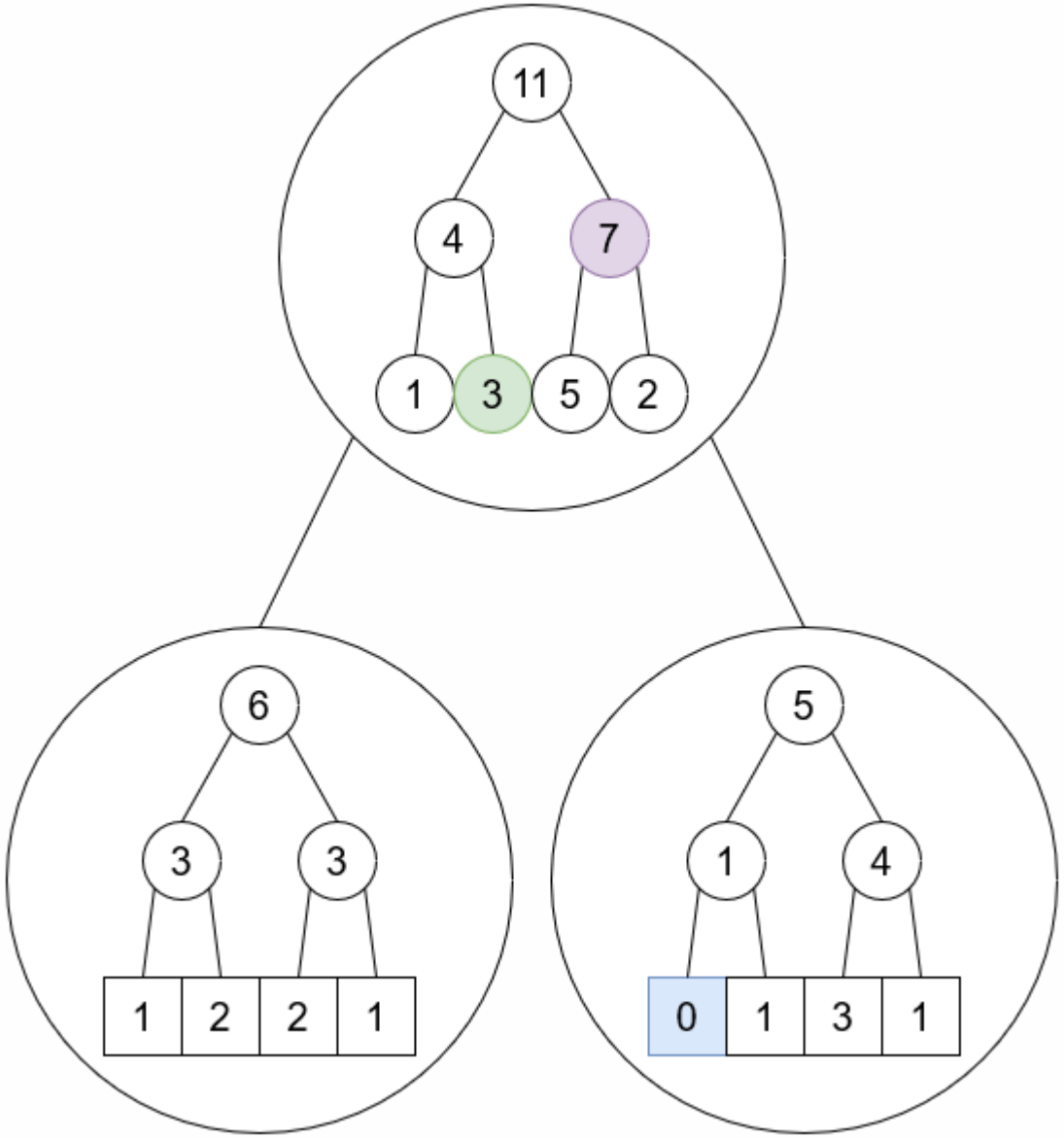
11	4	7	1	3	5	2
6	3	3	1	2	2	1
5	1	4	0	1	3	1

Многомерное ДО

- RSQ($a_1, b_1 \dots a_k, b_k$)
- Пусть $[l_i, r_i]$ – отрезок, за который отвечает вершина v размерности i
- Три типа вершин
  - $[\mathrm{l_i}, \mathrm{r}_i] \in [a_i, b_i]$
    - i = k: Вернуть значение
    - i ≠ k: Перейти на следующую размерность
  - $[\mathrm{l_i}, \mathrm{r}_i] \cap [a_i, b_i] = \emptyset$ – вернуть нейтральный элемент
  - $[\mathrm{l_i}, \mathrm{r}_i] \cap [a_i, b_i] \neq \emptyset$ – пойти к детям

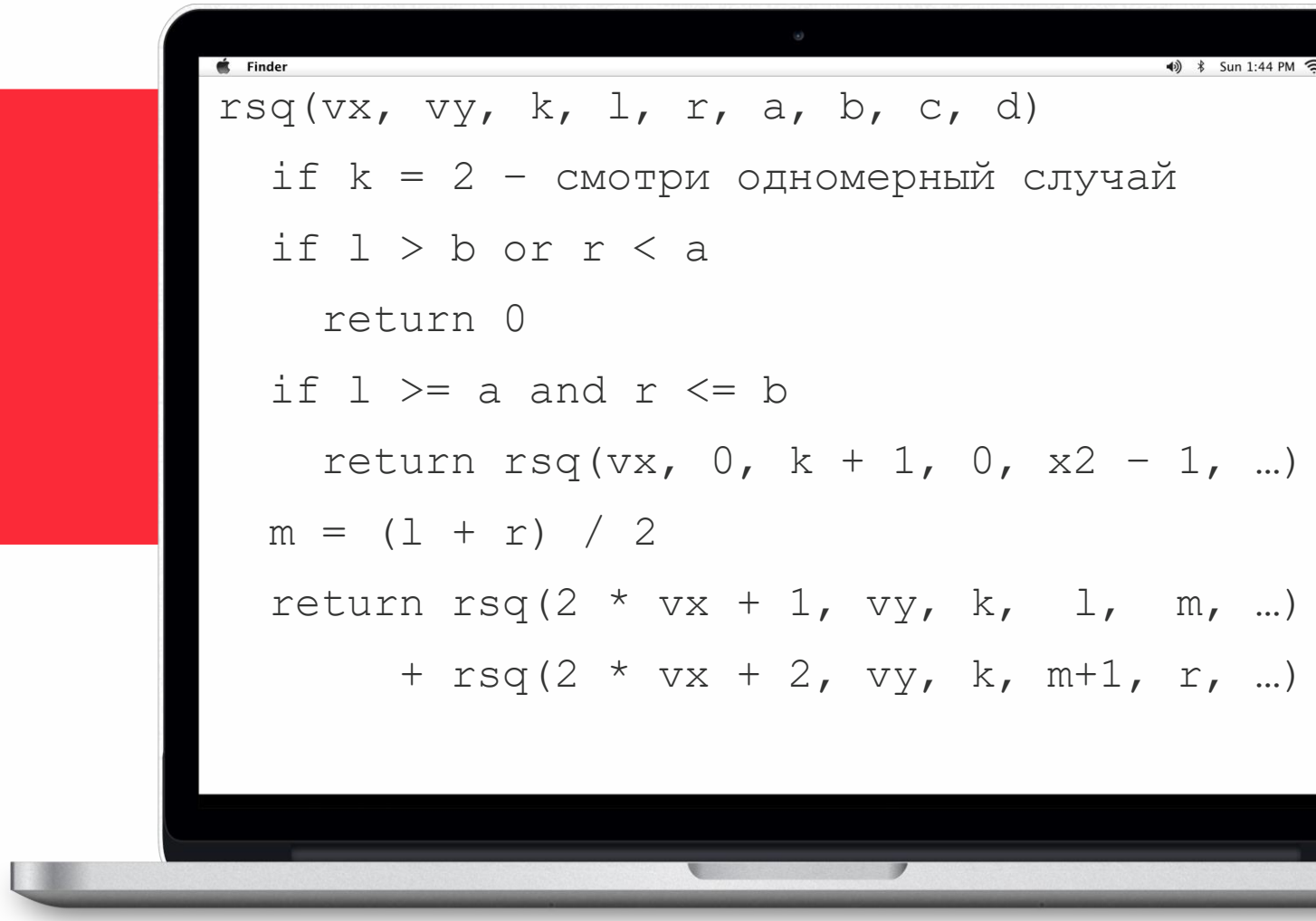
Многомерное ДО

1	2	2	1
0	1	3	1



Многомерное ДО

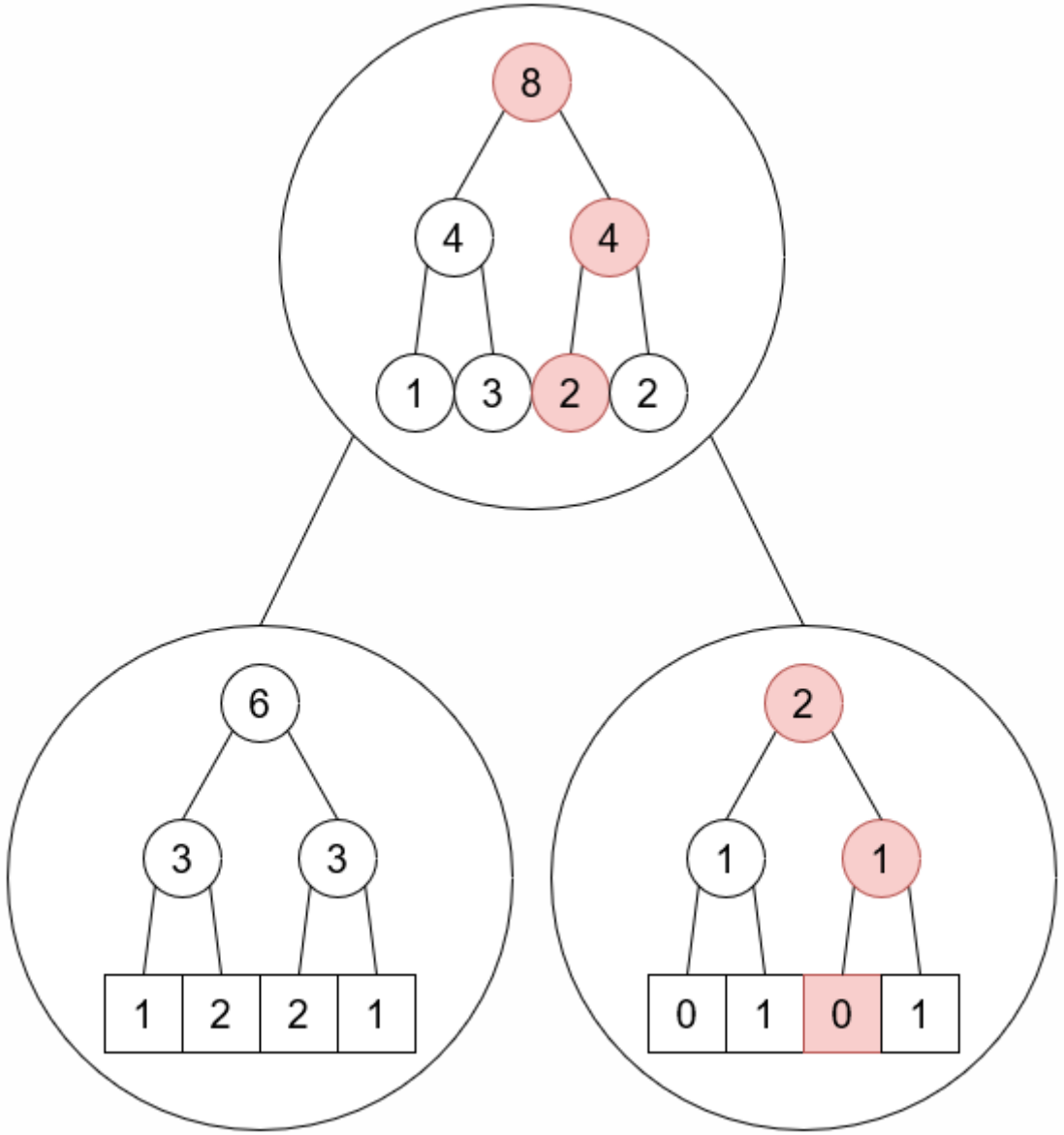
RSQ(a, b, c, d)



Многомерное ДО

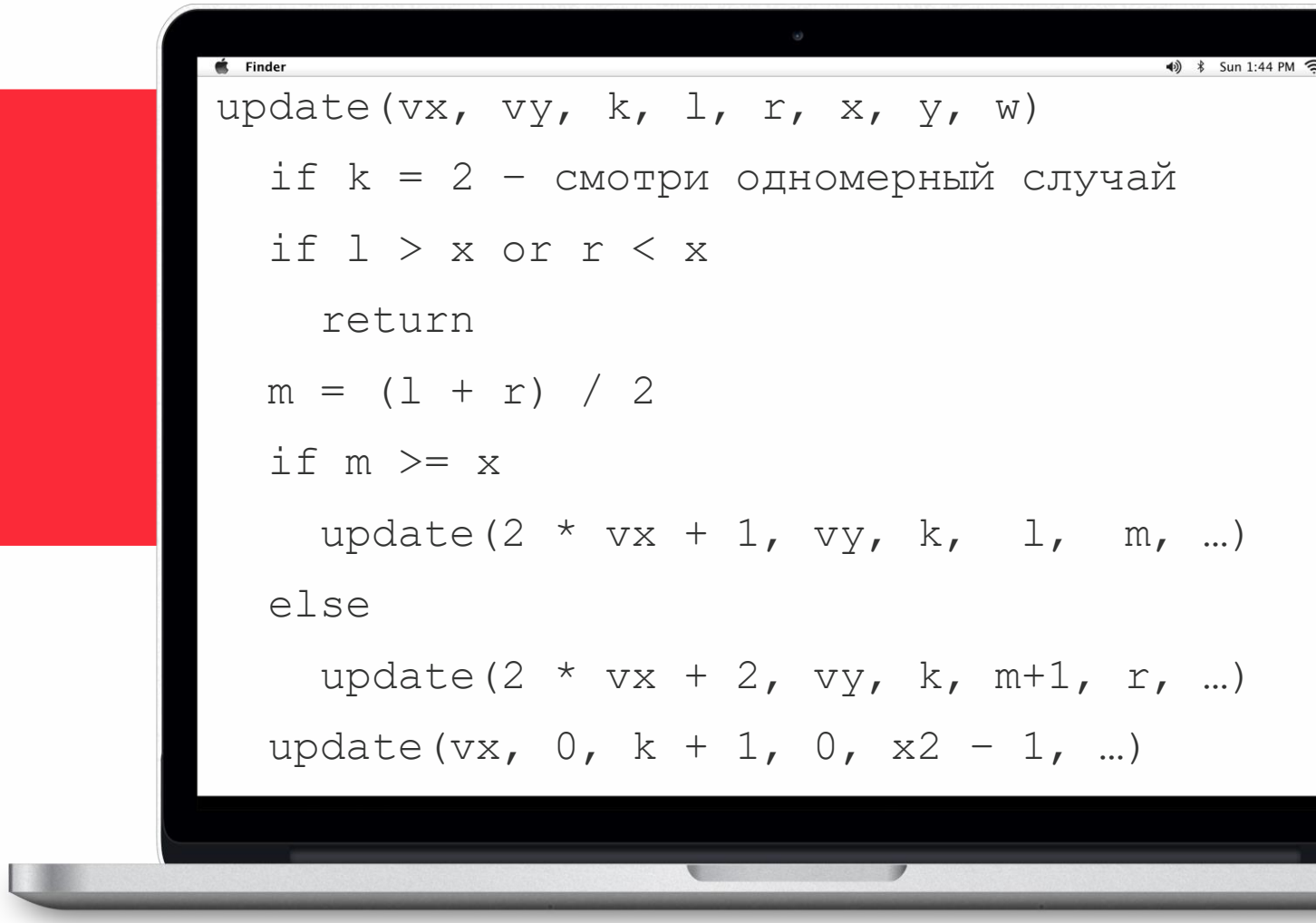
- Обновление в точке
- a[1][2] = 0

1	2	2	1
0	1	0	1



Многомерное ДО

RSQ(a, b, c, d)



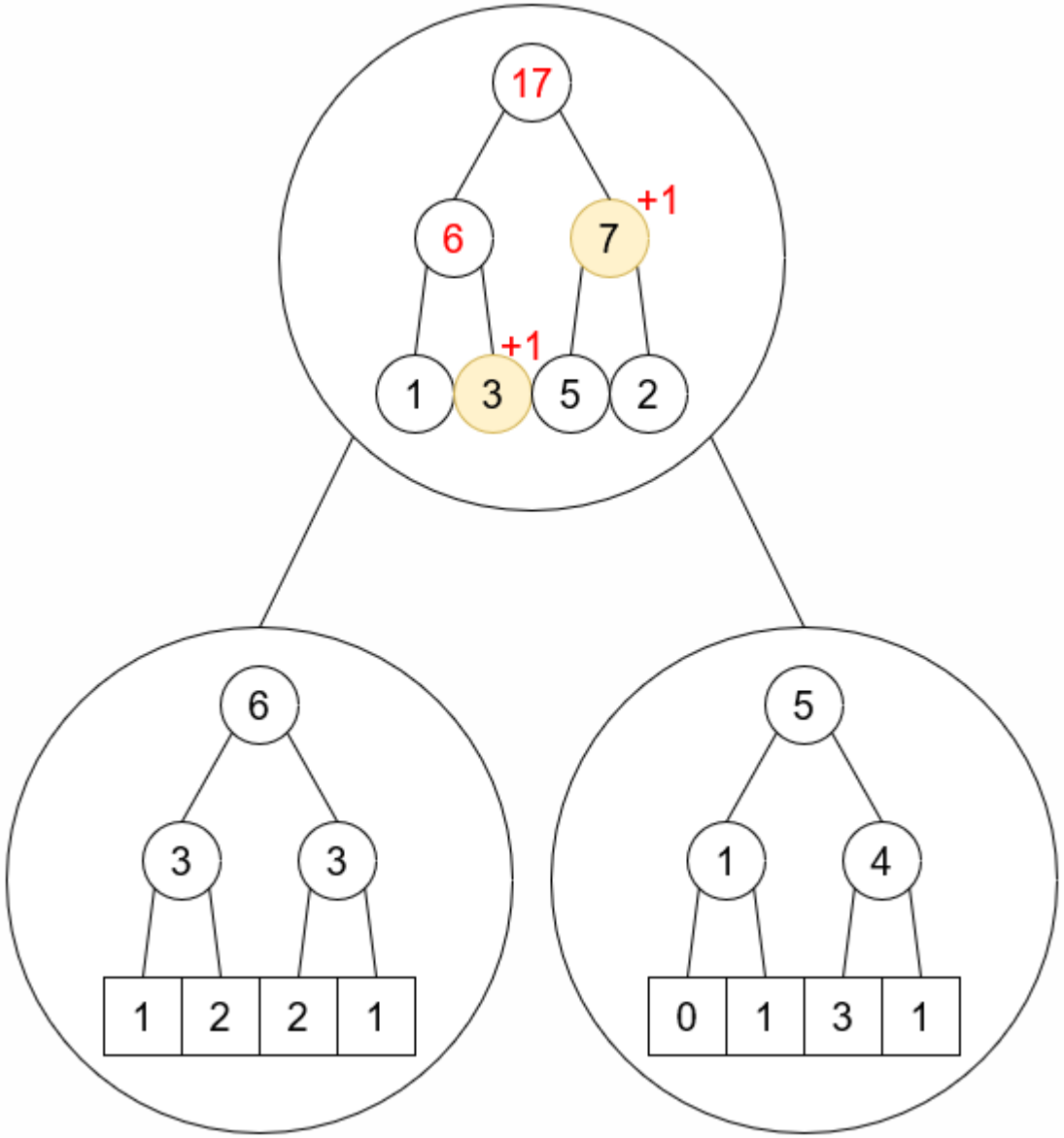
Многомерное ДО

- Обновление на прямоугольнике

+1

1	2	2	1
0	1	3	1

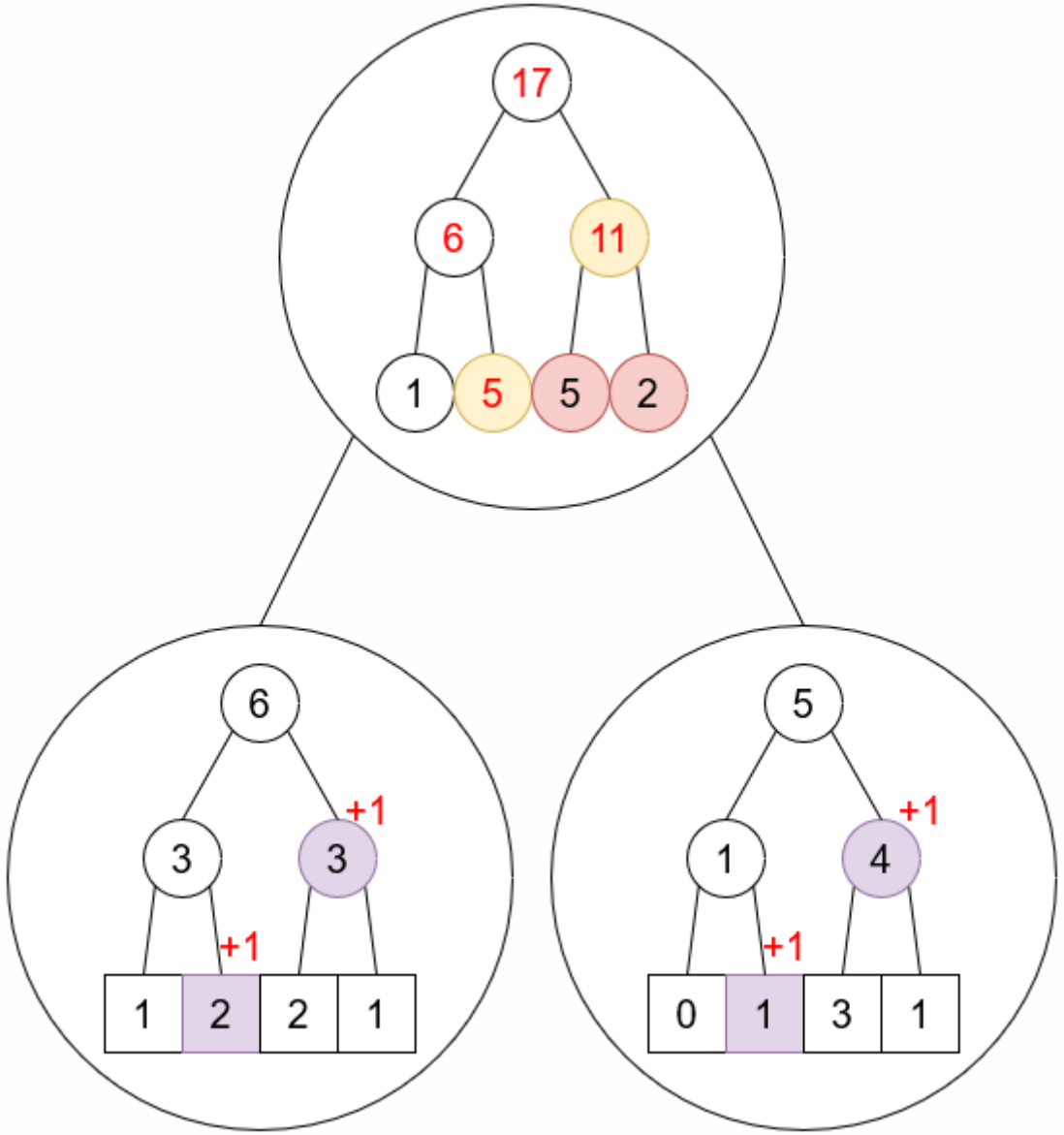
1	3	3	2
0	2	4	2



Многомерное ДО

- Проталкивать сложно
- Пометки хранить тоже в многомерном ДО!

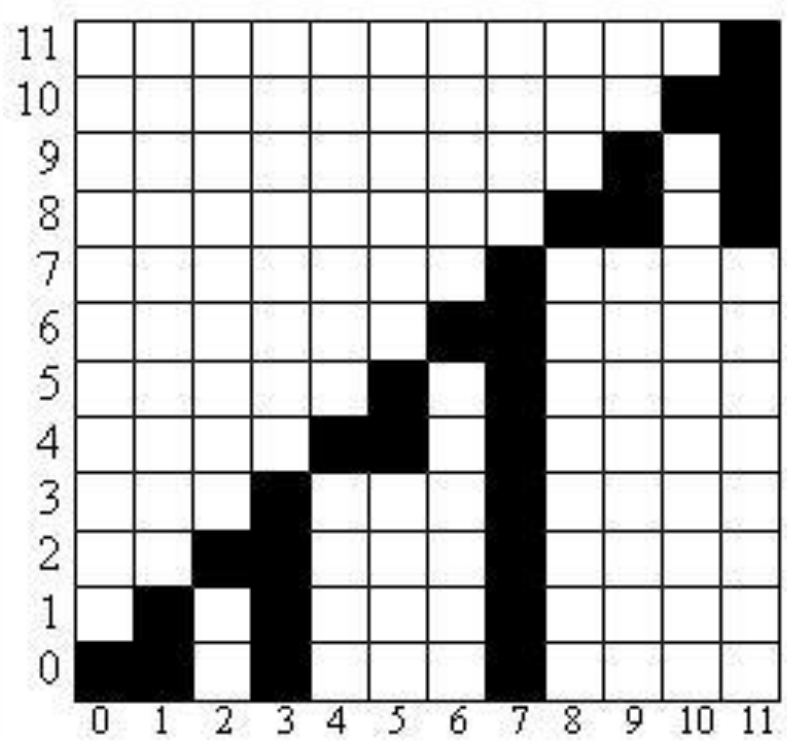
- Считаем сумму пометок на отрезке, домножаем на размер пересечения!



Многомерное Дерево Фенвика

Многомерное Дерево Фенвика

- $T[i][j] = \sum_{i'=F(i)}^{i} \sum_{j'=F(j)}^{j} a[i'][j']$
- F(i) = i & (i+1)

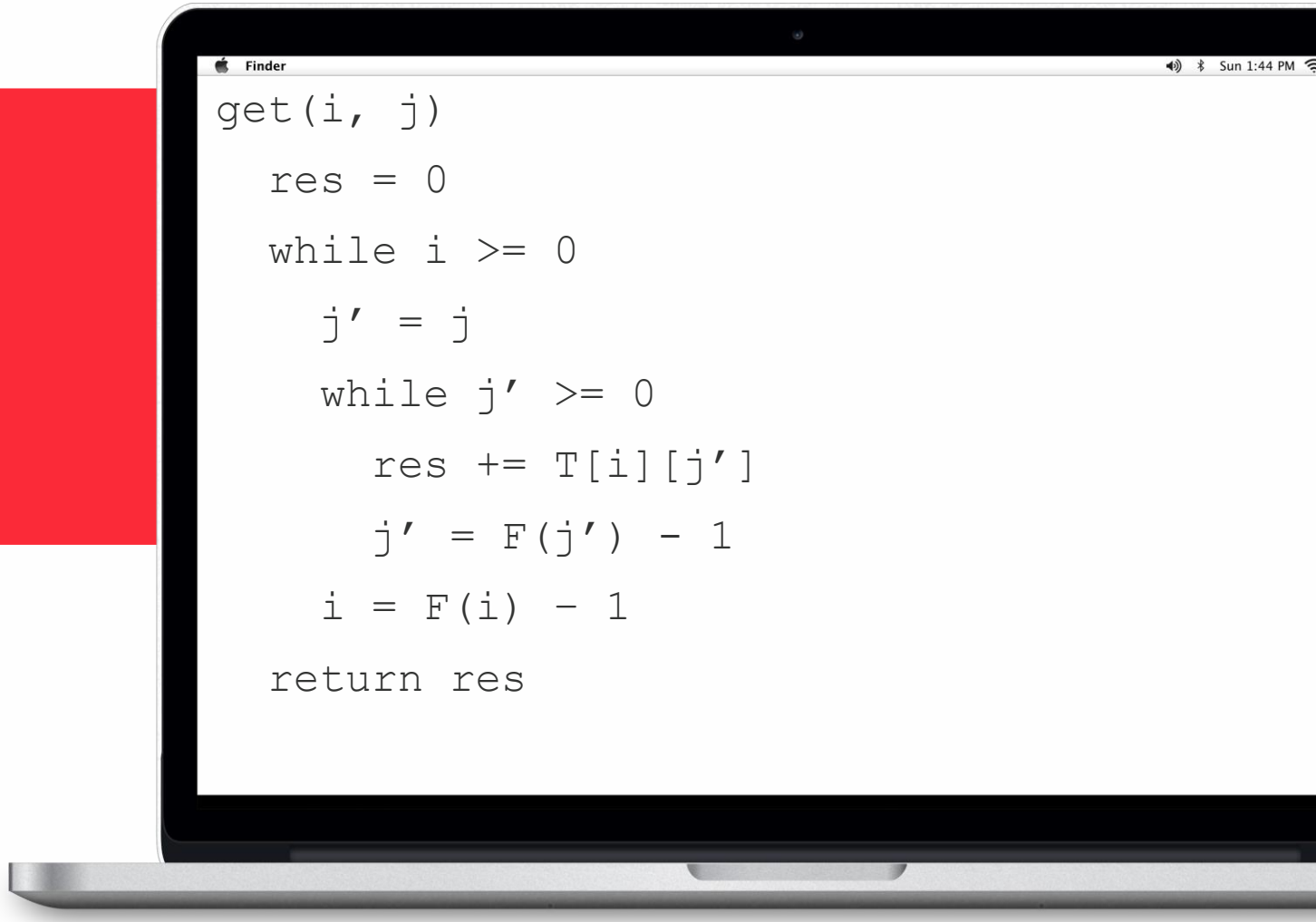


Многомерное Дерево Фенвика

- Запрос rsq(l1, r1, l2, r2)
- Научимся считать префиксные суммы
  - Сведем задачу к уже решенной!

Многомерное Дерево Фенвика

Запросы суммы

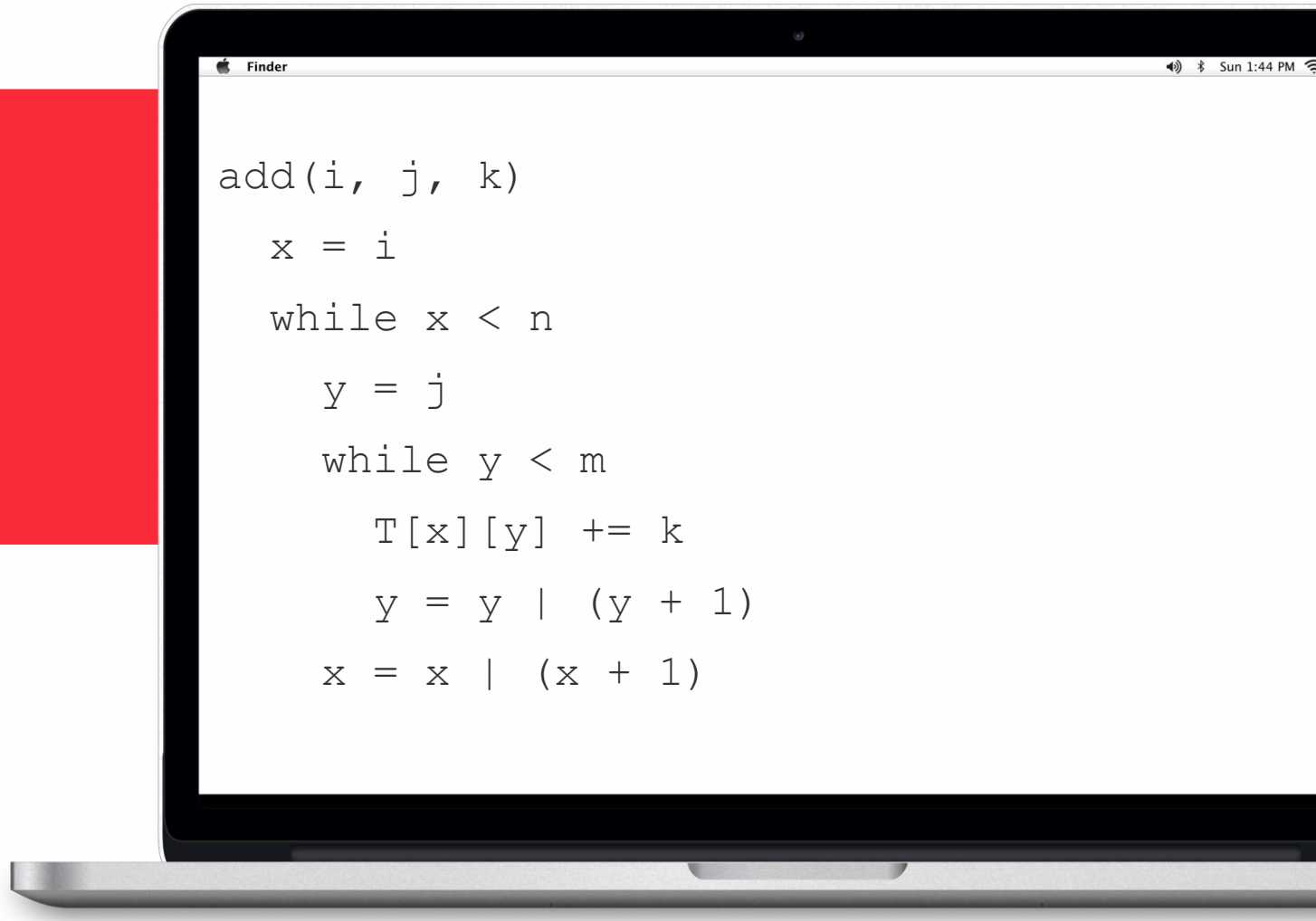


Многомерное Дерево Фенвика

- Запрос add(i, x)
- Аналогично!

Многомерное Дерево Фенвика

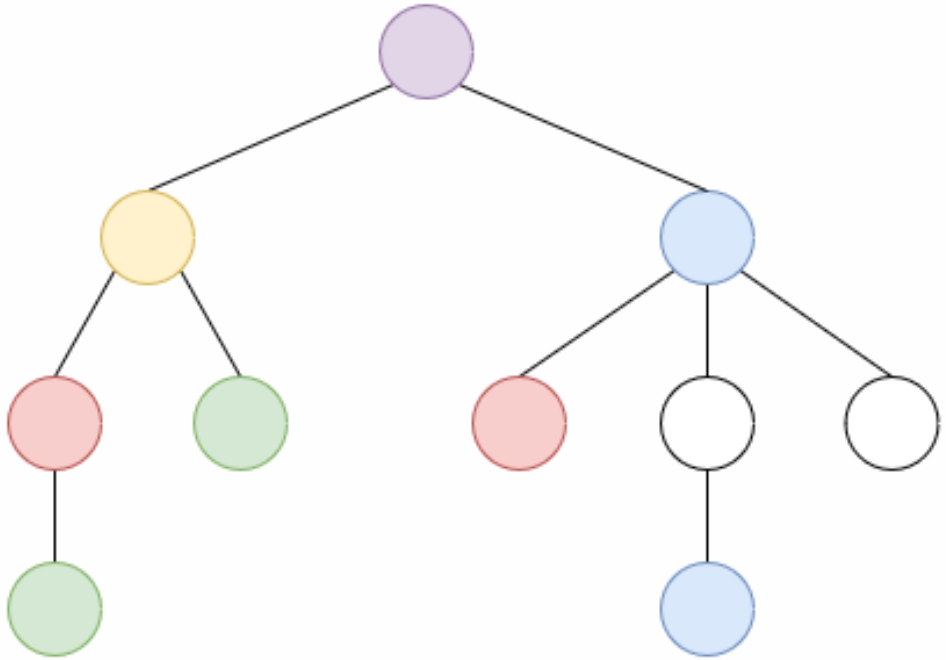
Запросы add(i, j, k)



LCA

LCA

- LCA – Lowest Common Ancestor (наименьший общий предок)
- «Точка перегиба» пути из u в v (в дереве путь один)



LCA

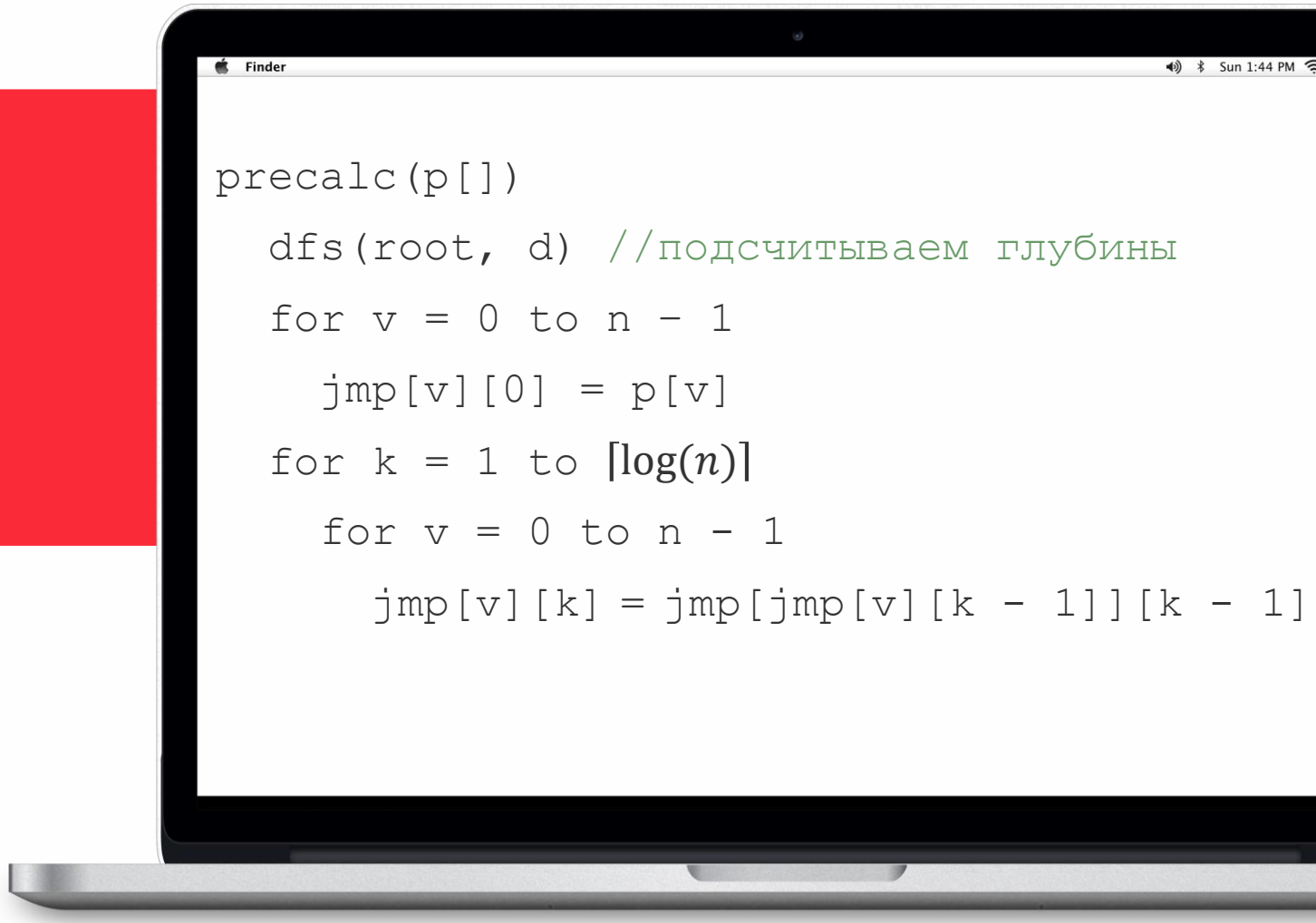
- Как искать?
  - Наивно
  - Случай одинаковой глубины
  - Что делать, если разные?

LCA

- Как ускорить?
- Двоичные подъемы
  - jmp[v][i] – куда попадем из v, через 2^i прыжков (nlogn значений)
  - jmp[v][0] = p[v]
  - jmp[v][i] = jmp[jmp[v][i-1]][i-1]

LCA

Двоичные подъемы:
предподсчет

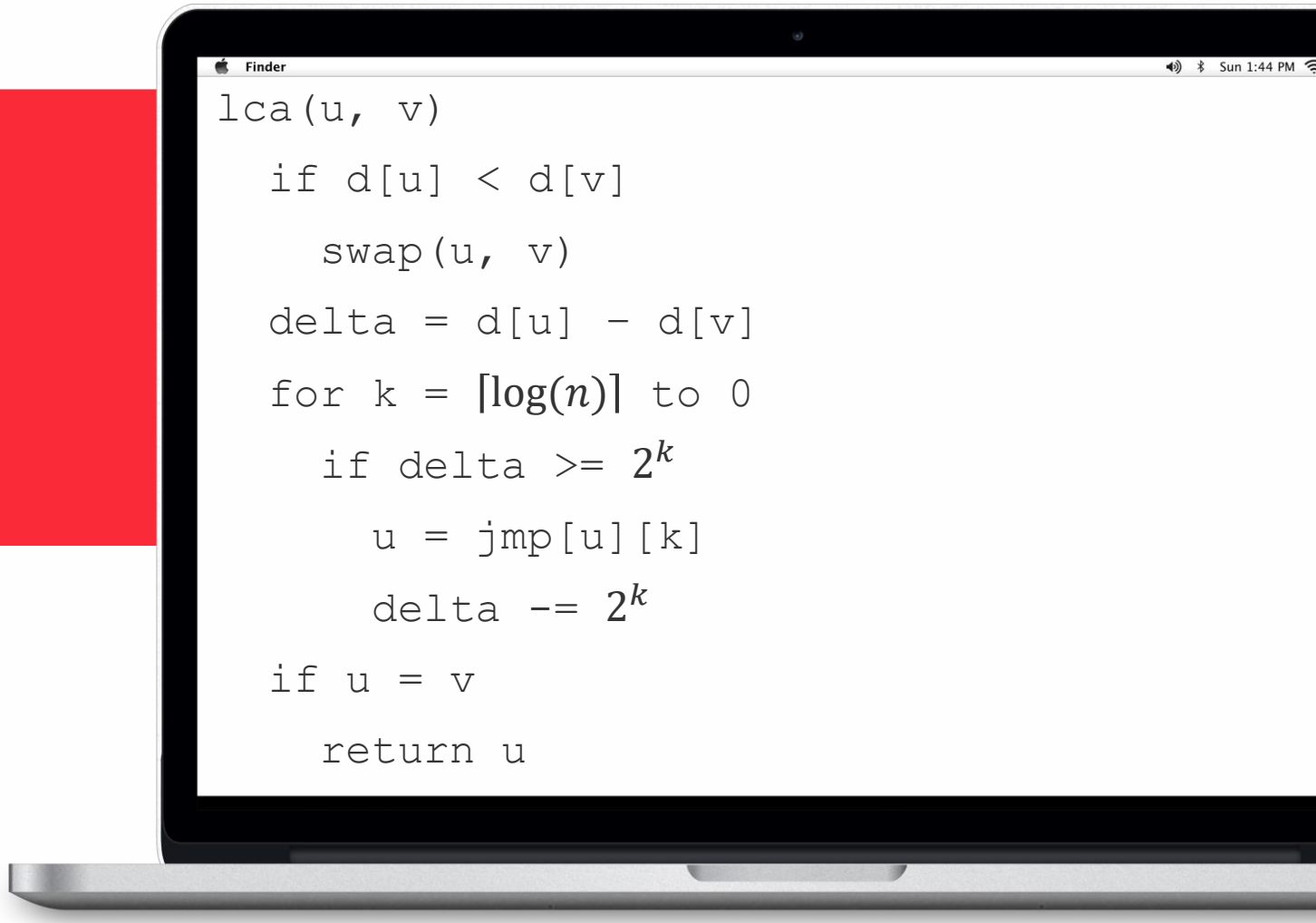


LCA

- Запрос LCA(u, v)
- Поднимаемся на одну высоту:
  - Более глубокую вверх на |d[u] - d[v]|
    - O(logn)
- Найдем такой наибольший прыжок h, что, если подняться из u и v на такую высоту будем в разных вершинах
  - O(logn)

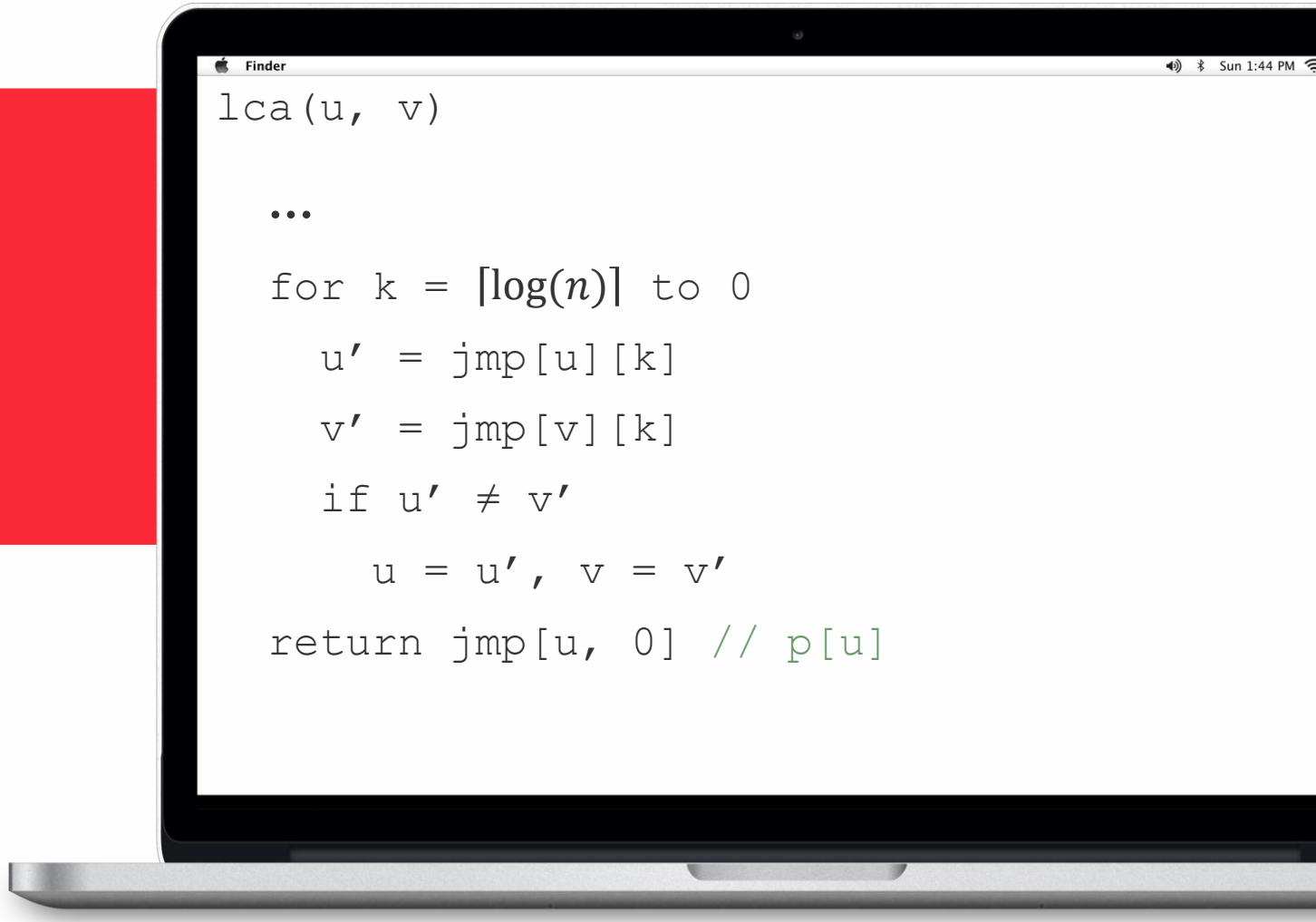
LCA

Двоичные подъемы: запрос, подъем на одну высоту



LCA

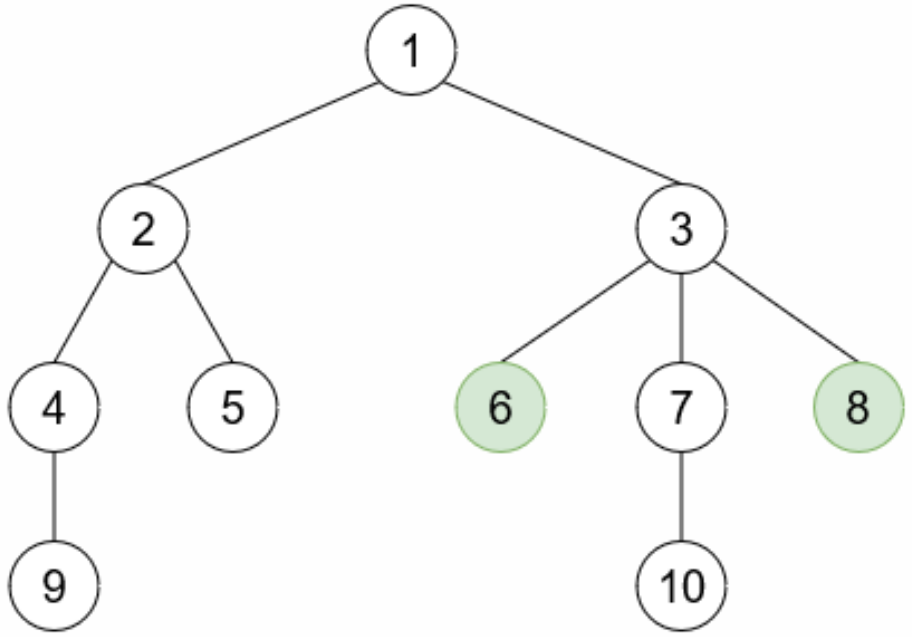
Двоичные подъемы: запрос



Сведение LCA к RMQ

Сведение LCA к RMQ

- Эйлеров обход дерева
  - LCA лежит на отрезке между u и v
  - Имеет минимальную высоту
- RMQ по глубинам



1	2	4	9	4	2	5	2	1	3	6	3	7	10	7	3	8	3	1

0	1	2	3	2	1	2	1	0	1	2	1	2	3	2	1	2	1	0

Сведение LCA к RMQ

	Двоичные подъемы	Дерево отрезков	Sparse Table
Предподсчет	O(nlogn)	O(n)	O(nlogn)
Запрос	O(logn)	O(logn)	O(1)

Все!