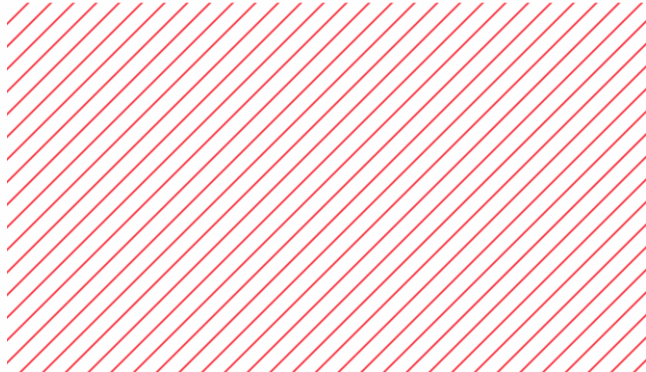
академия больших данных

mail.ru group

Запросы на отрезках

Шовкопляс Григорий

Введение в алгоритмы и структуры данных



Что такое запросы на отрезках?

Запросы на отрезках

- Есть массив чисел, к нему можно последовательно применять запросы:
  - Сумма на отрезке с L по R: $\sum_{i=L}^{R} a[i]$
  - Минимум/максимум на отрезке с L по R: $min/max_{i=L}^{R} a[i]$
  - Изменить i-й элемент массива: $a[i] = x$
  - Прибавить x на отрезке с L по R: $\text{for i in [L; R]: } a[i] \mathrel{+}= x$
  - И другие…

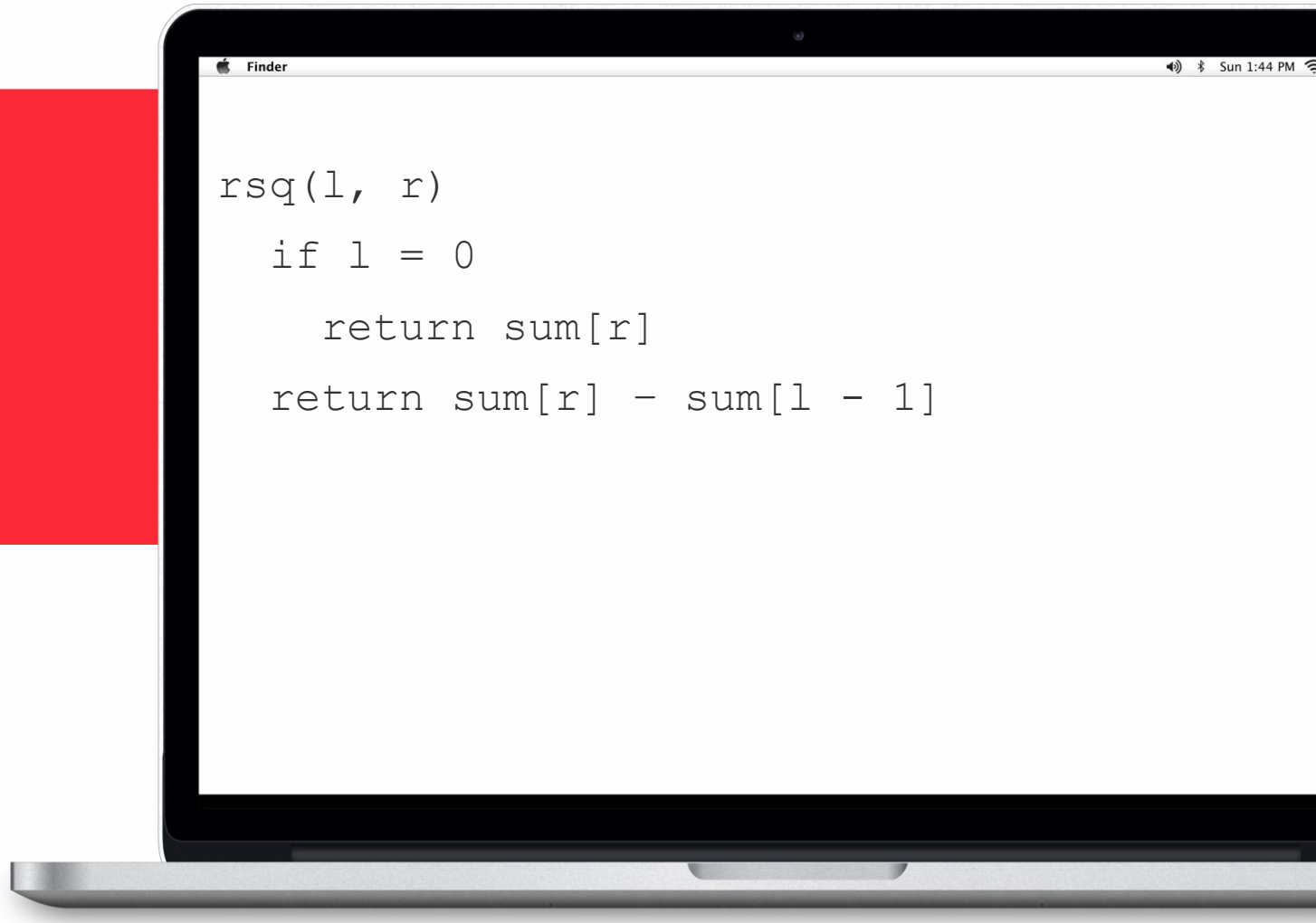
Префиксные суммы

Префиксные суммы

- Сможем за *O(1)* отвечать на запрос *RSQ(l, r)*
- Пусть дан массив *a[i]*
- Посчитаем массив *sum[i]* = $\sum_{j=0}^{i} a[i]$
  - Как считать?
  - $sum[i] = sum[i-1] + a[i]$
- Чему равно *RSQ(l, r)*?
  - $sum[r] - sum[l-1]$
  - $l = 0$?
    - $sum[r]$!

Префиксные суммы

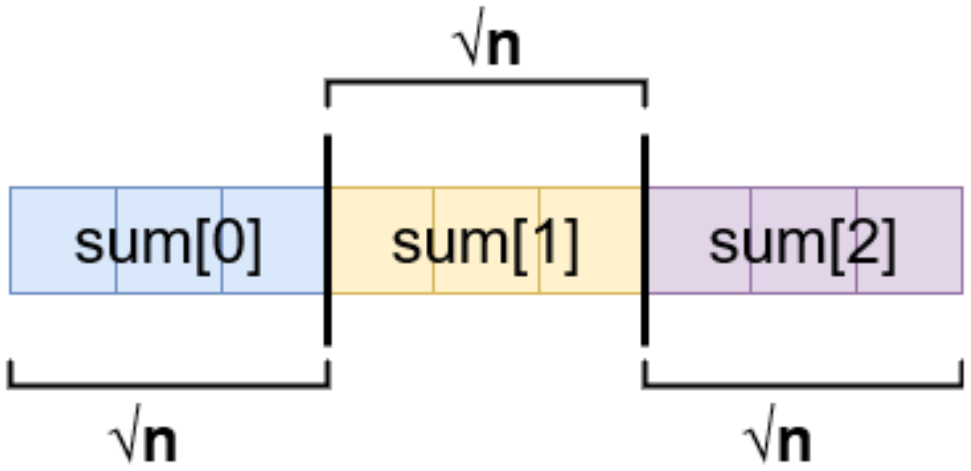
Как писать?



Корневая эвристика (Sqrt decomposition)

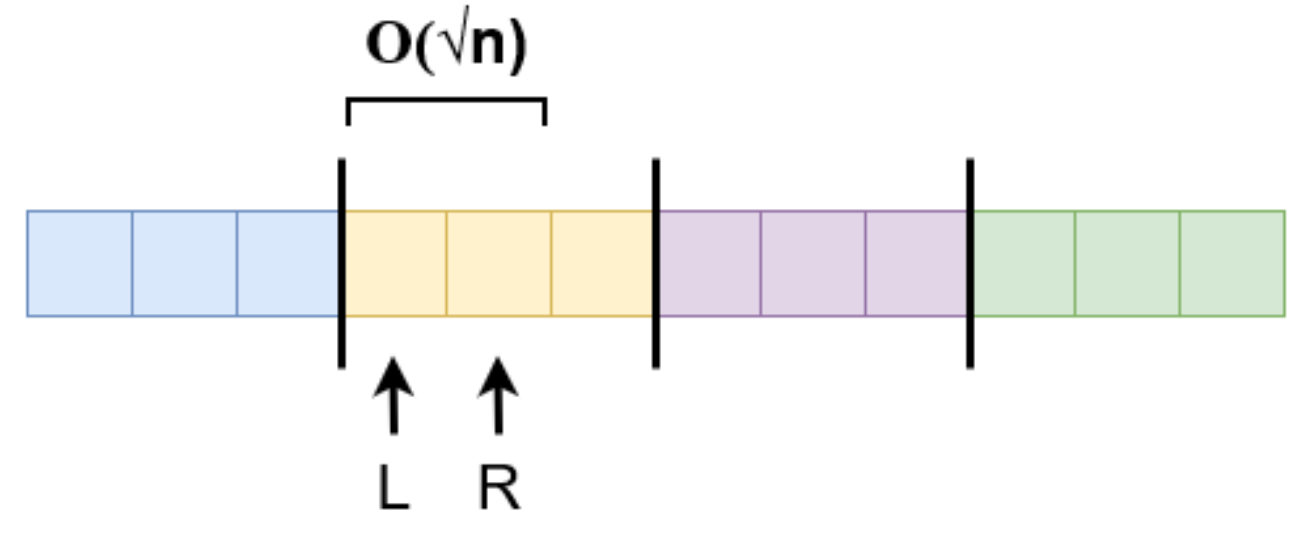
Корневая эвристика

- Есть ассоциативная операция (сумма/максимум/минимум)
  - $a + (b + c) = (a + b) + c$
- Отвечаем на запрос на отрезке за $O(\sqrt{n})$



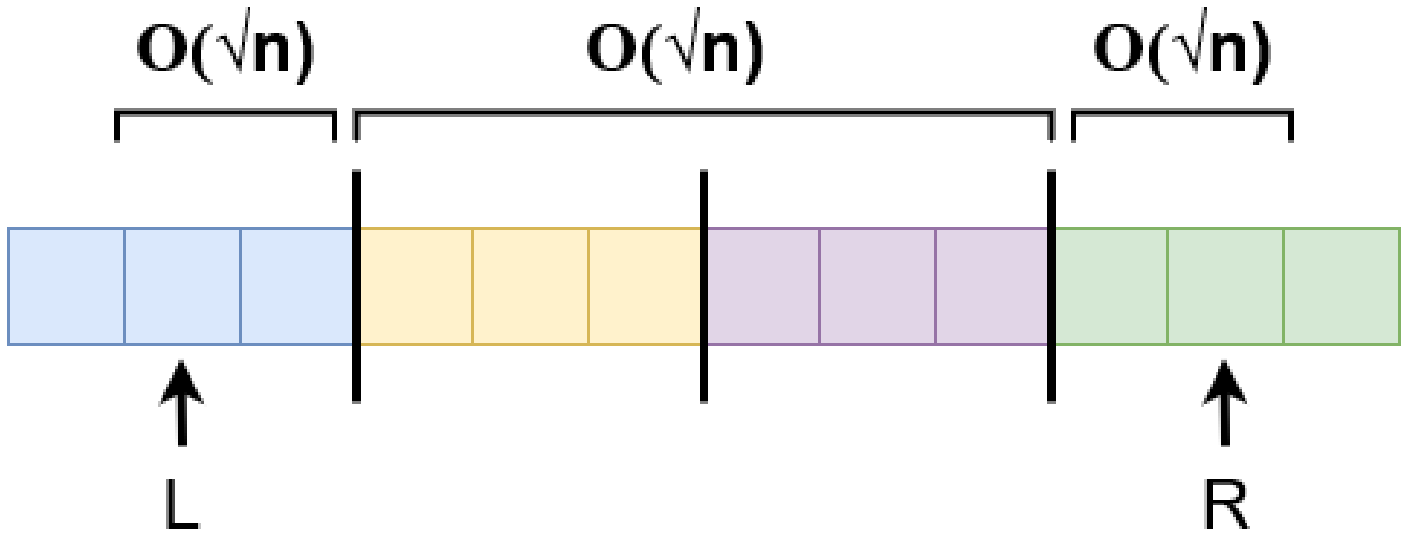
Корневая эвристика

- Отвечаем на запрос: L и R внутри одного блока
- $\left\lfloor \frac{L}{\sqrt{n}} \right\rfloor = \left\lfloor \frac{R}{\sqrt{n}} \right\rfloor$



Корневая эвристика

- Отвечаем на запрос: L и R в разных блоках



Разреженная таблица (Sparse Table)

Разреженная таблица

- Хотим отвечать на запросы *RMQ* за *O(1)*
- Префиксные суммы не работают для минимума/максимума
  - Минимум не обратим
- Наивная идея:
  - Для всех l, r храним: min[l][r] – $min_{i=l}^{r}\, a[i]$
  - *O(1)* запрос
  - *O(n^3)* предподсчет

Разреженная таблица

- Улучшим предподсчет $O(n^3) \to O(n^2)$
- Динамическое программирование:
  - min[i][i] = a[i]
  - min[l][r] = min(min[l][r-1], a[r])

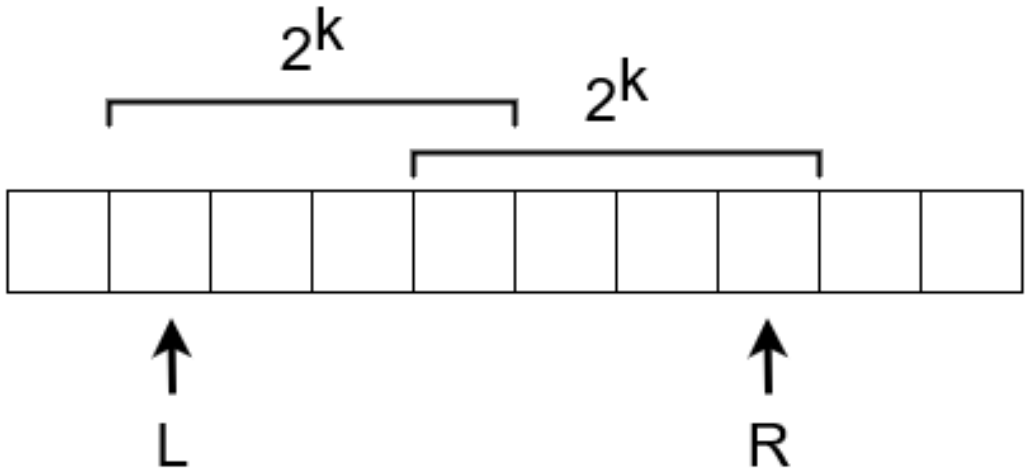
- Идемпотентность: min(a, a) = a

Разреженная таблица

- Улучшим предподсчет и память $O(n^2) \to O(nlogn)$
- min[l][k] – $min_{i=l}^{l+2^k-1} a[i]$: минимум на отрезке $[l; l+2^k)$
  - Размер таблицы $O(nlogn)$
- Как считать?
- Динамическое программирование:
  - min[i][0] = a[i]
  - min[l][k] = min(min[l][k-1], min[l+2^{k-1}][k-1])
  - Предподсчет $O(nlogn)$

Разреженная таблица

- Как отвечать на запрос за *O(1)*?
- $k: \max\{i \mid 2^i \leq r - l + 1\}$
- rmq(l, r) = min(min[i][k], min[?][k])
- rmq(l, r) = min(min[i][k], min[r−2^k+1][k])



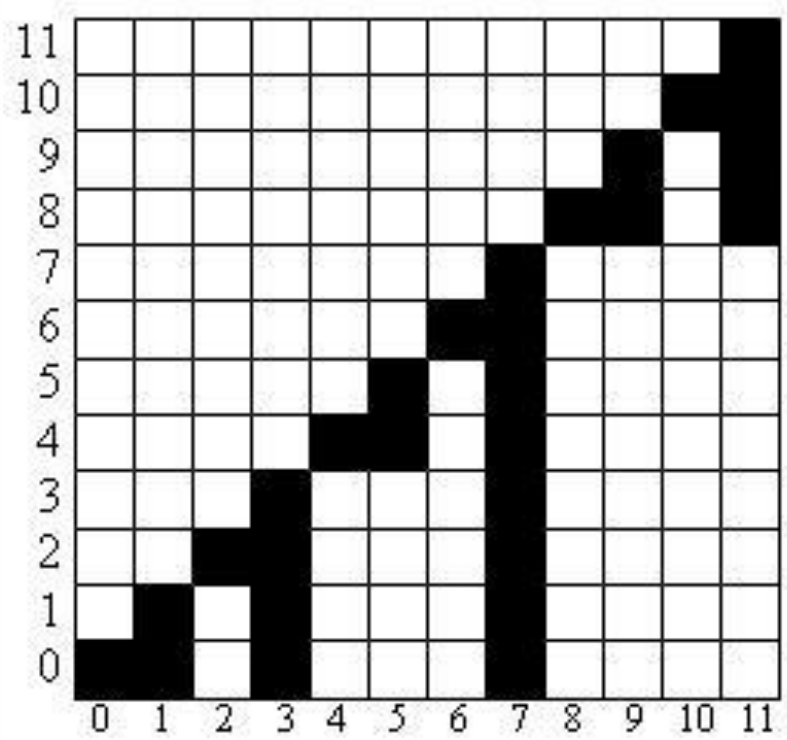
Дерево Фенвика

Дерево Фенвика

- Есть три вида запросов:
  - rsq(l, r)
  - set(i, x)
  - add(i, x)

Дерево Фенвика

- Обобщим префиксные суммы:
  - $T[i] = \sum_{j=F(i)}^{i} a[j]$
- В дереве Фенвика F(i) = i & (i+1)
- На картинке заштрихованы элементы массива, которые входят в сумму T[i]

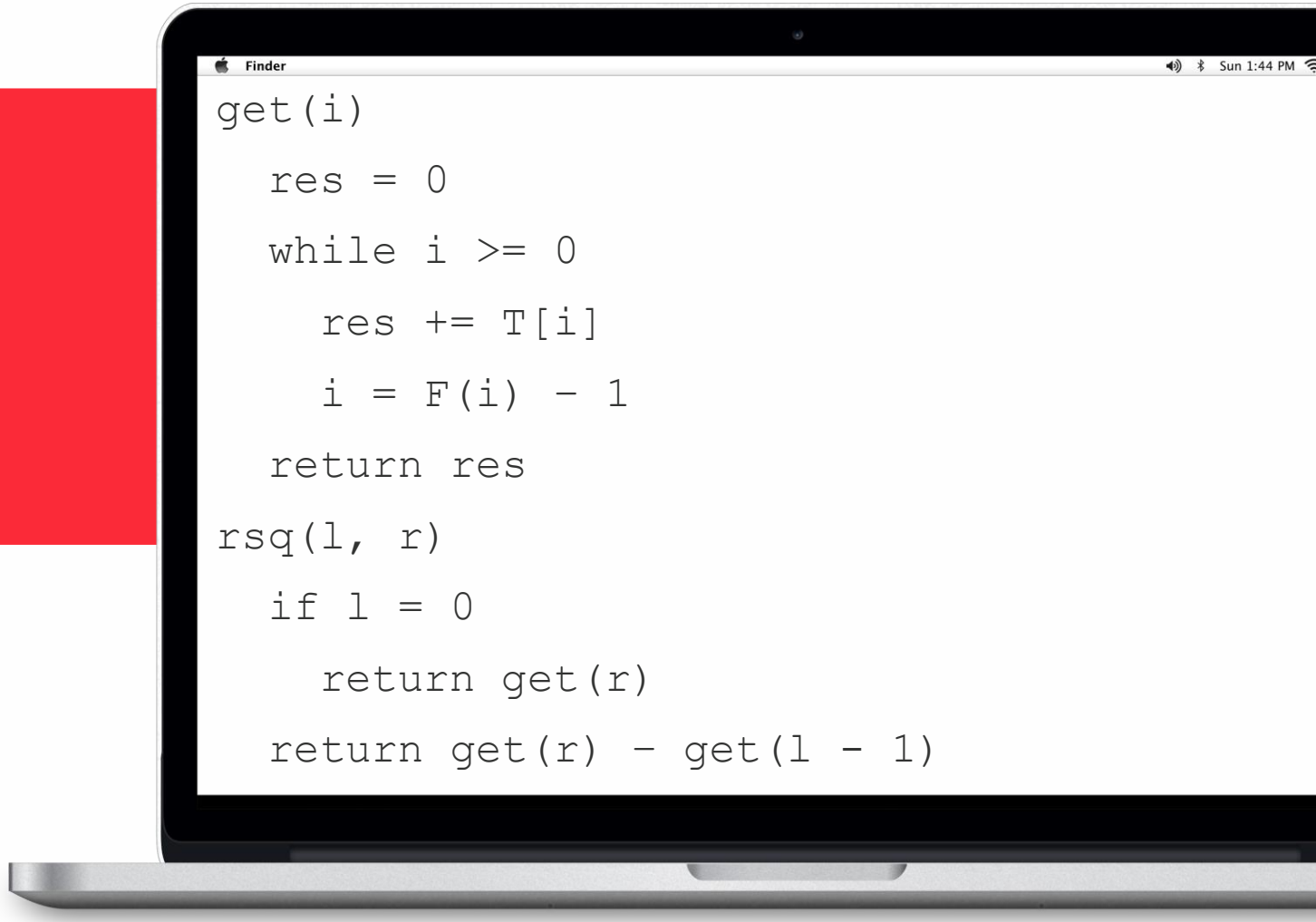


Дерево Фенвика

- Запрос rsq(l, r)
- Научимся считать префиксные суммы
  - Научимся считать сумму на отрезке
- sum[i] = T[i] + sum[F(i) - 1]
- Утверждение: sum[i] раскладывается в O(log i) элементов T[j]

Дерево Фенвика

Запросы суммы

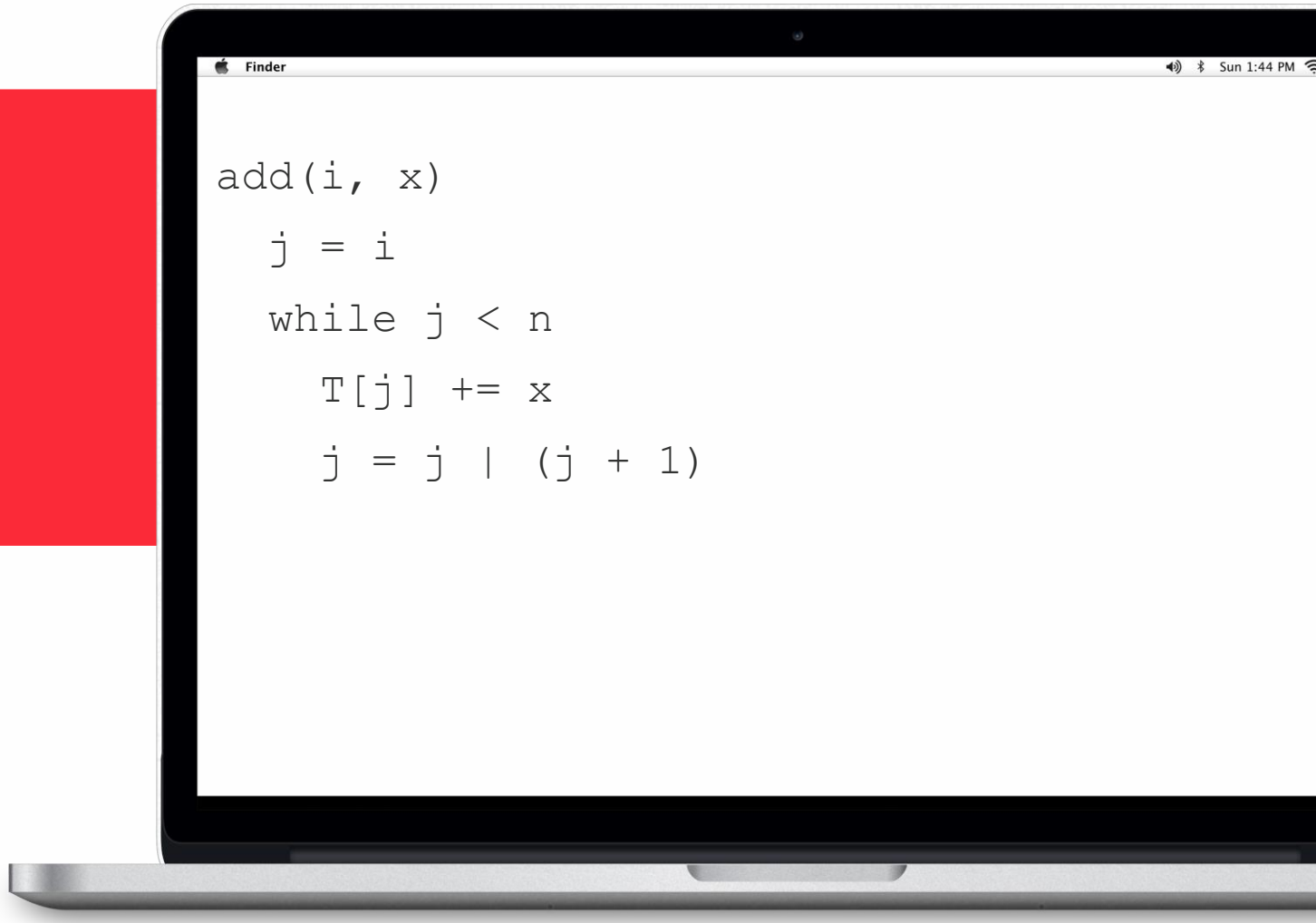


Дерево Фенвика

- Запрос add(i, x)
- Утверждение: a[i] входит в O(log n) элементов T[j]
- $F(j) \leq i \leq j$
- j можно вычислить по рекуррентной формуле:
  - $j_0 = i$
  - $j_k = j_{k-1} | (j_{k-1} + 1)$

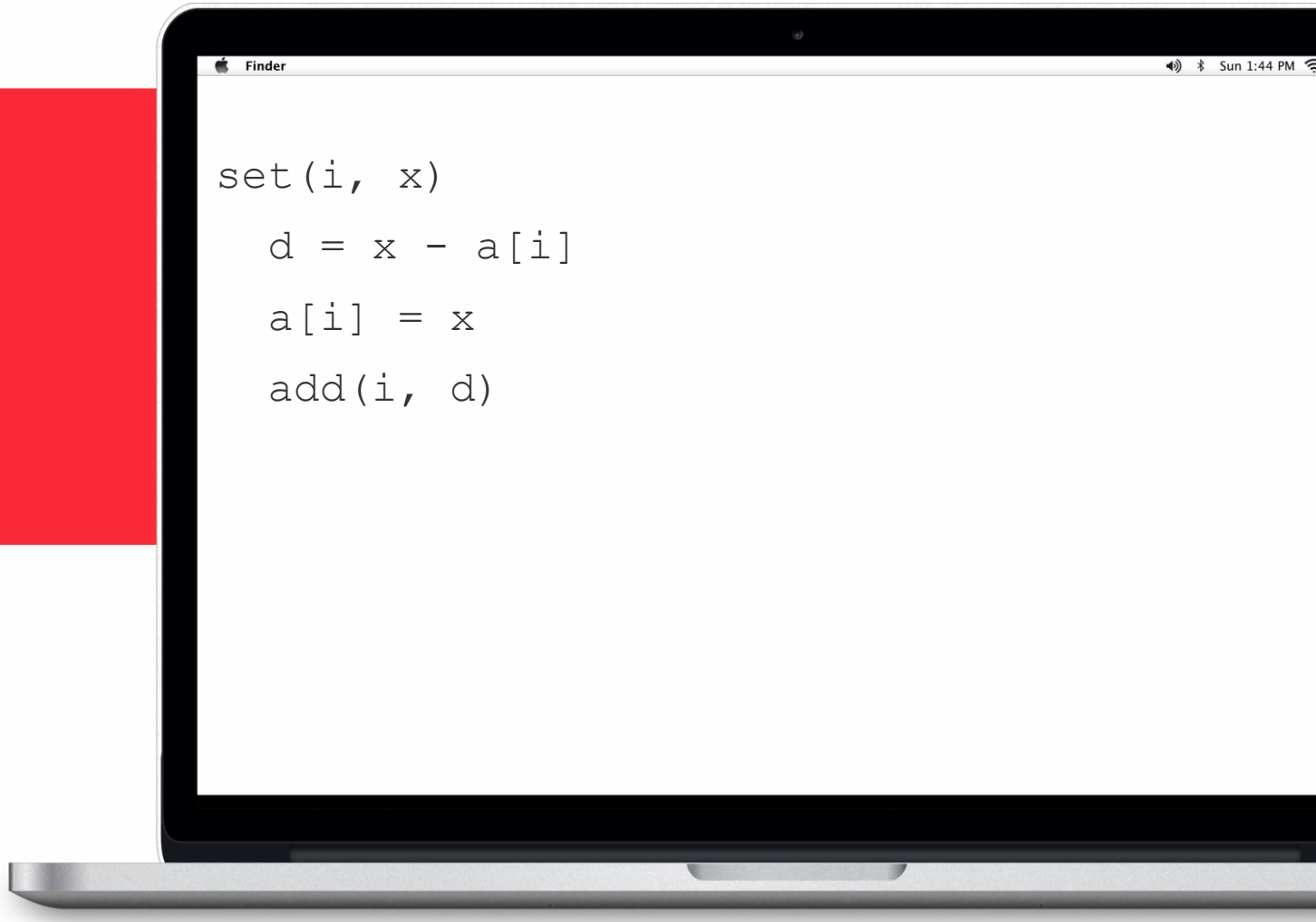
Дерево Фенвика

Запросы add(i, x)



Дерево Фенвика

Запросы set(i, x)

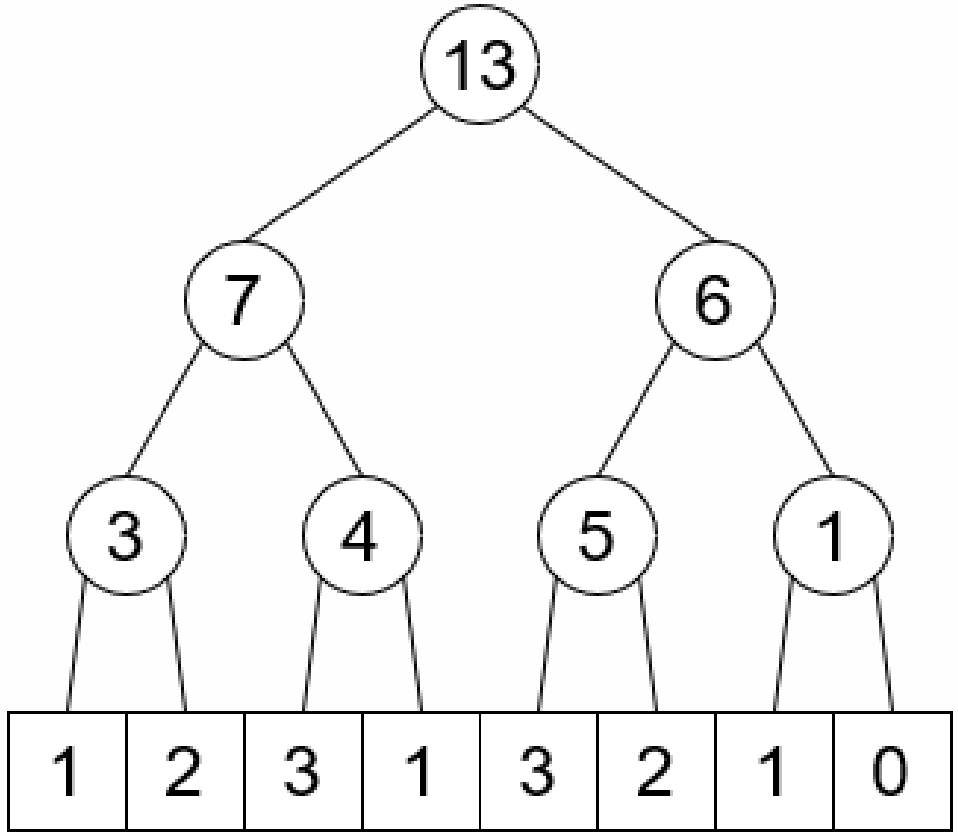


Дерево отрезков

Дерево отрезков

- Полное двоичное дерево
- Листы – элементы массива
- T[v] = T[2v + 1] + T[2v + 2]

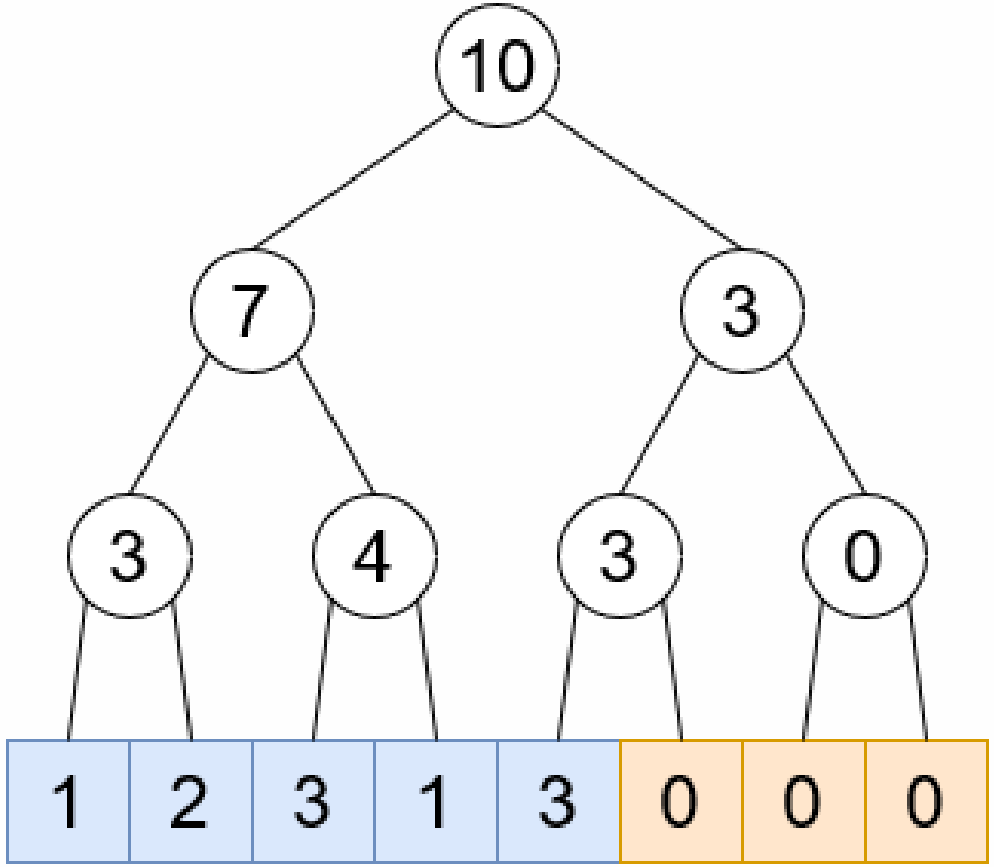
- Работает для любой ассоциативной операции



Дерево отрезков

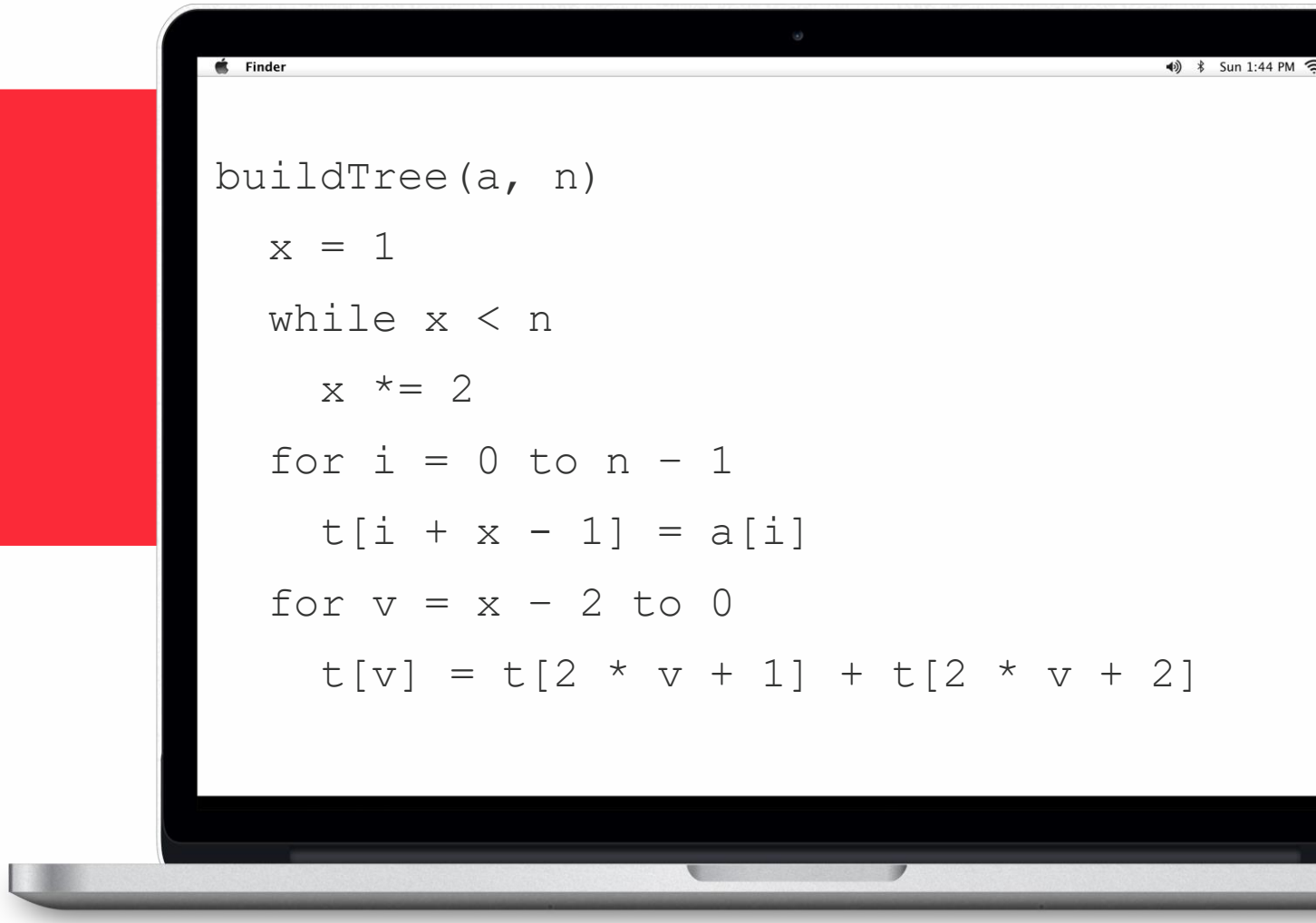
- Если длина массива не степень двойки?
- Нейтральный элемент

- $X = \min\{\, 2^k \mid 2^k \geq n\}$
- t[X-1+i] = a[i]



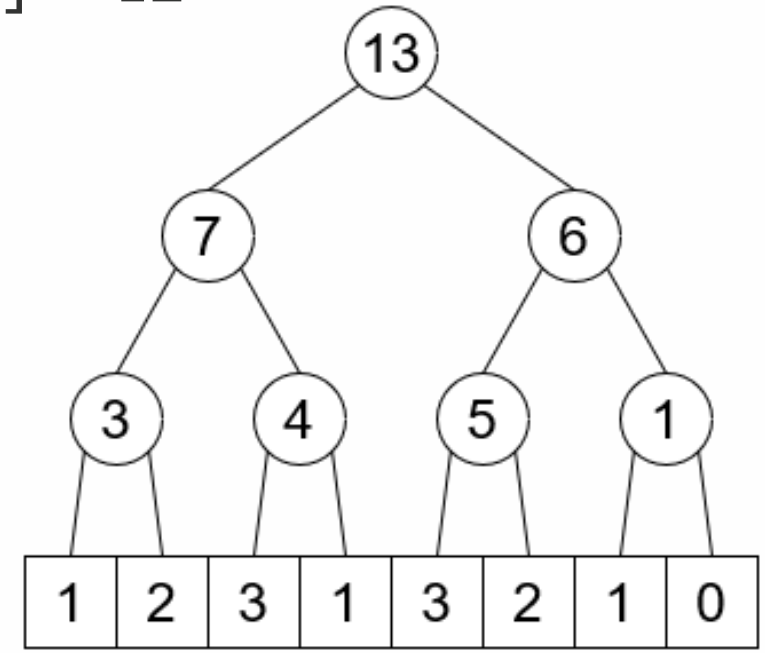
Дерево Отрезков

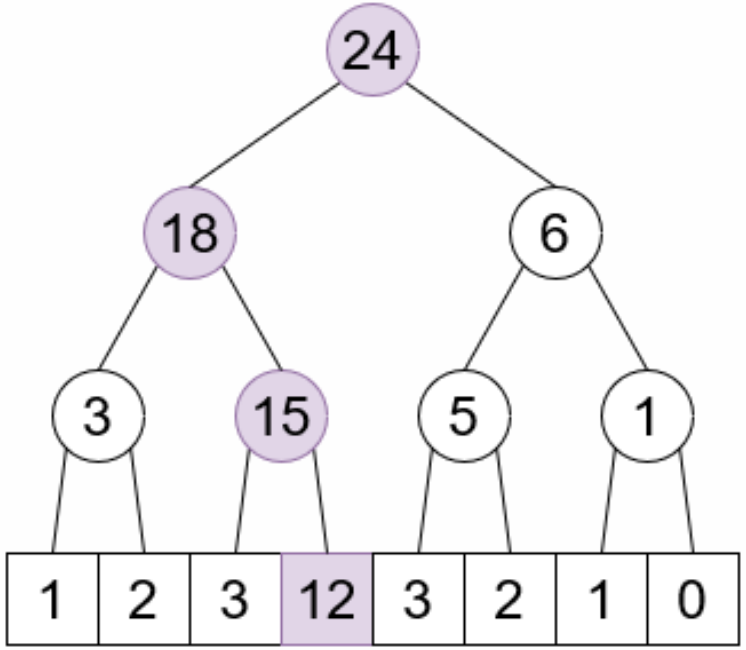
Построение



Дерево отрезков

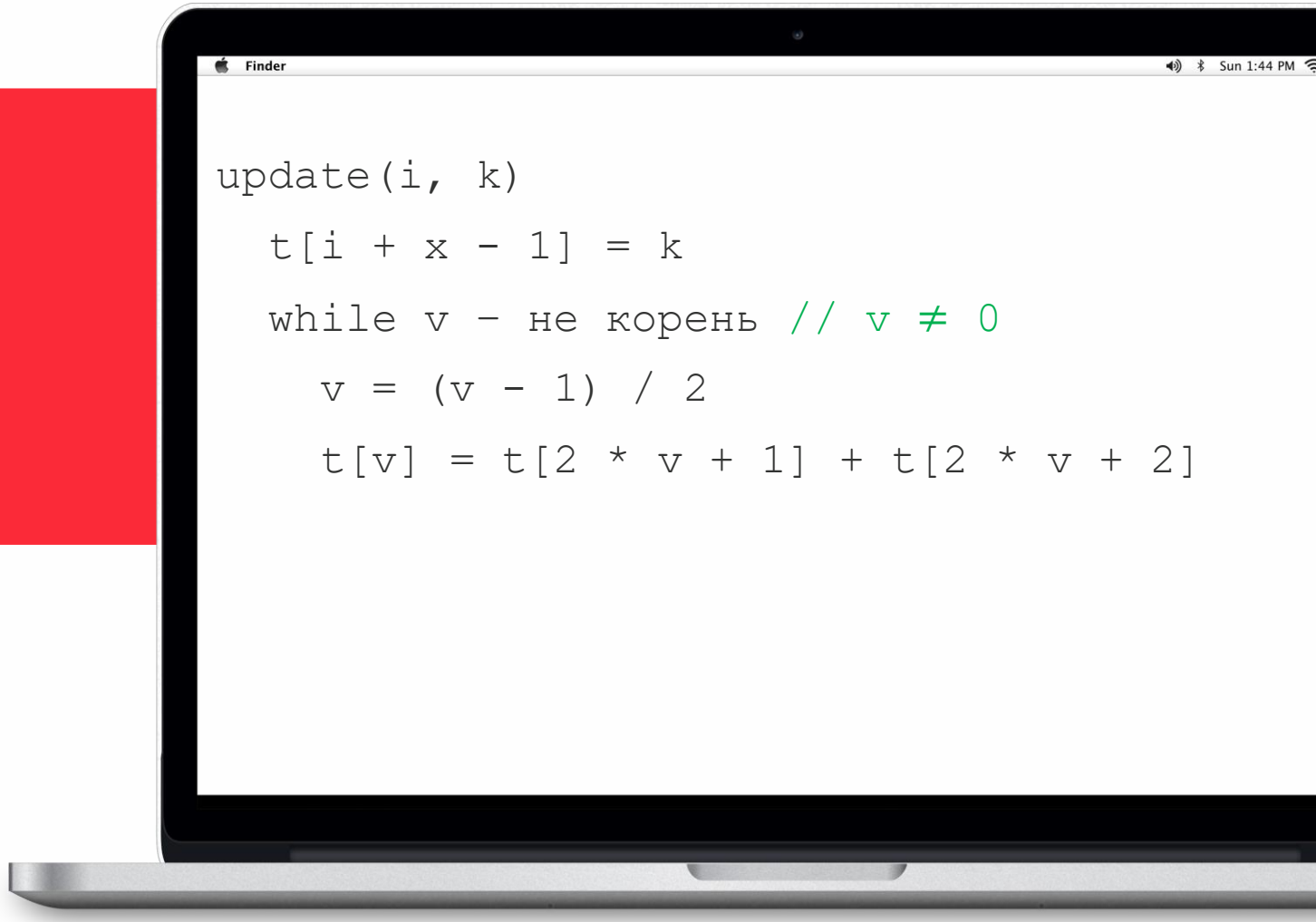
- Обновление в точке
- a[3] = 12





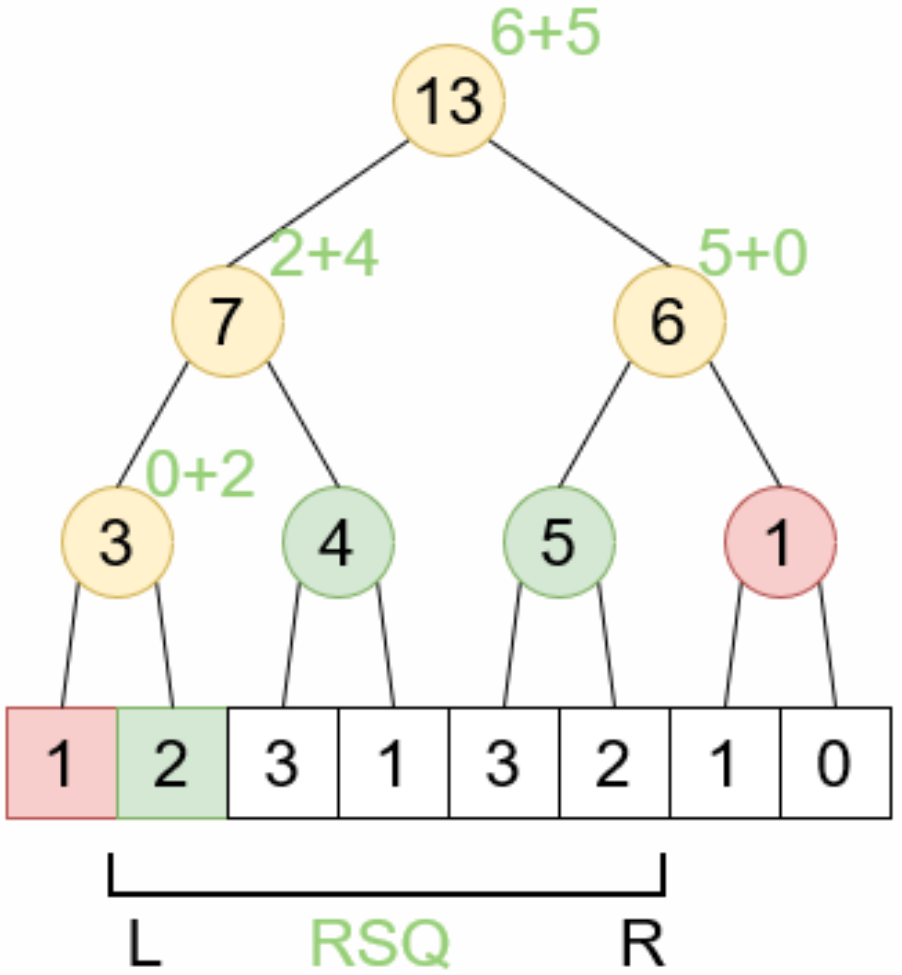
Дерево Отрезков

Обновление в точке



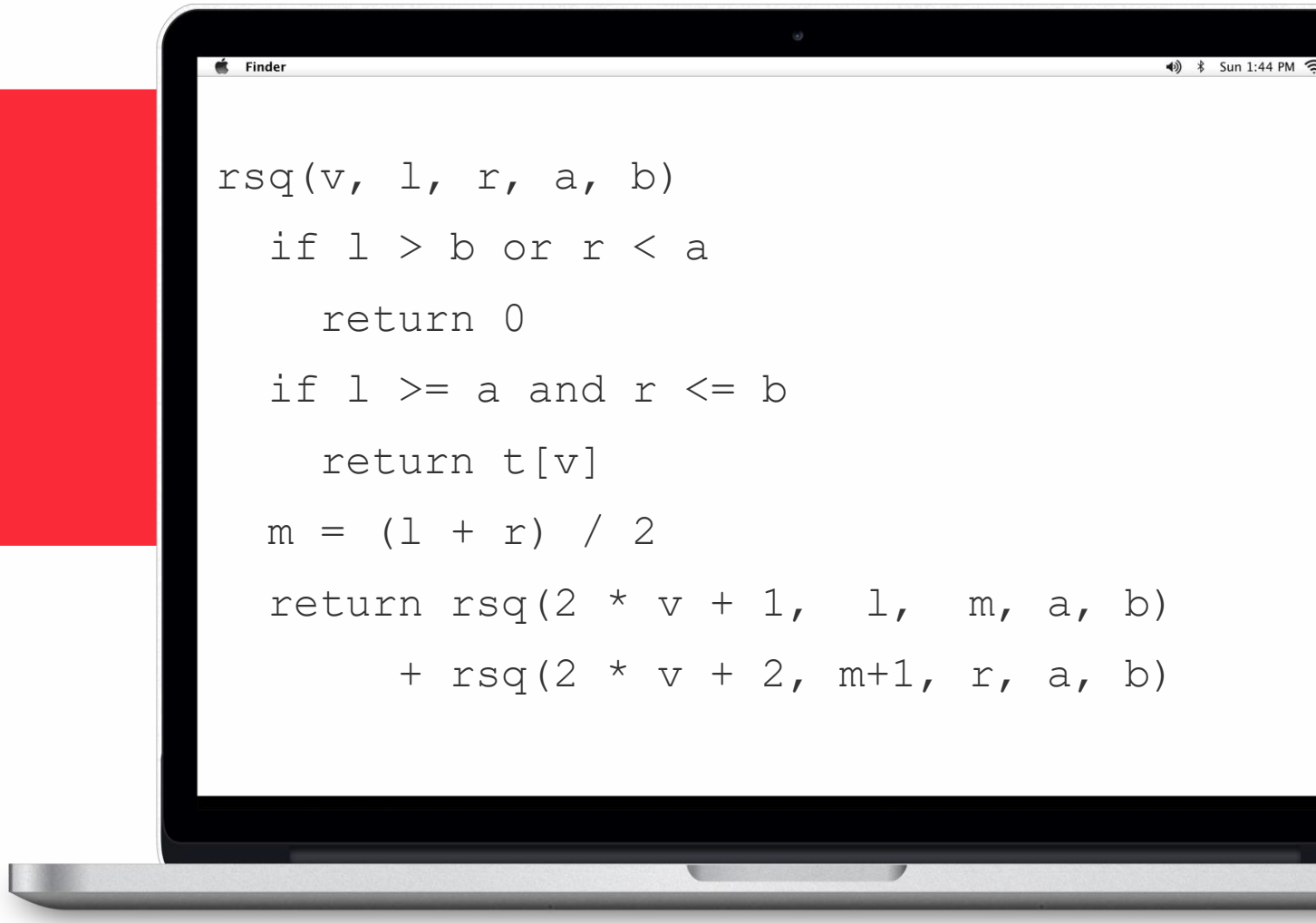
Дерево отрезков

- rsq(a, b)
- Пусть [l, r] – отрезок, за который отвечает вершина v
- Три типа вершин
  - $[\mathrm{l}, \mathrm{r}] \in [a, b]$
  - $[\mathrm{l}, \mathrm{r}] \cap [a, b] = \emptyset$
  - $[\mathrm{l}, \mathrm{r}] \cap [a, b] \neq \emptyset$



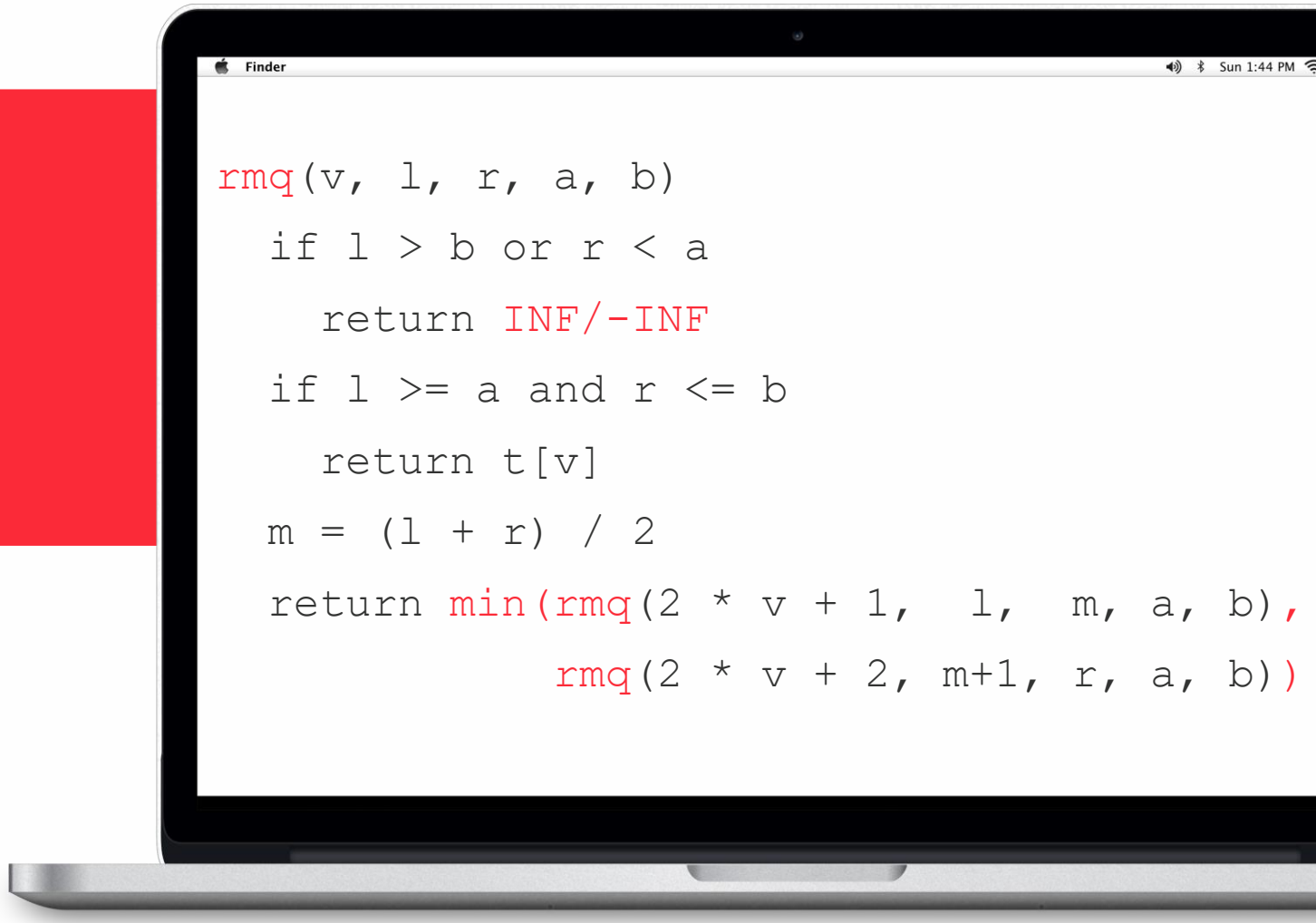
Дерево Отрезков

RSQ(a, b)



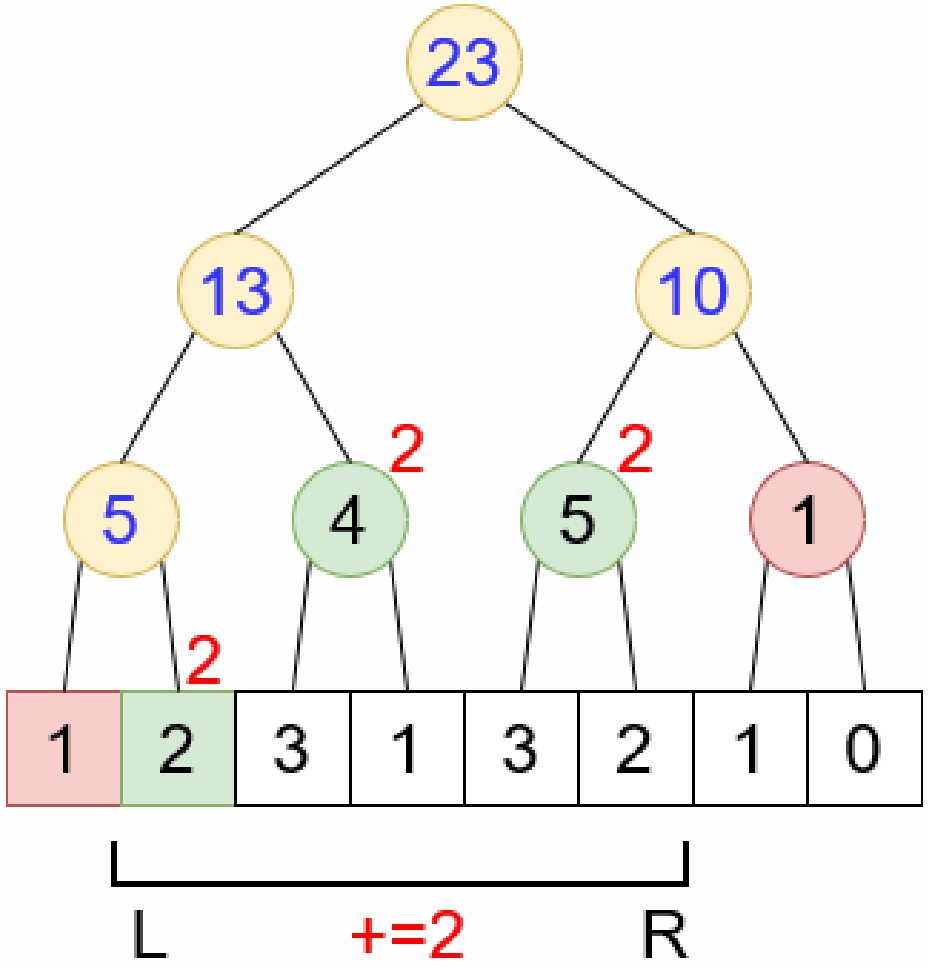
Дерево Отрезков

RMQ(a, b)



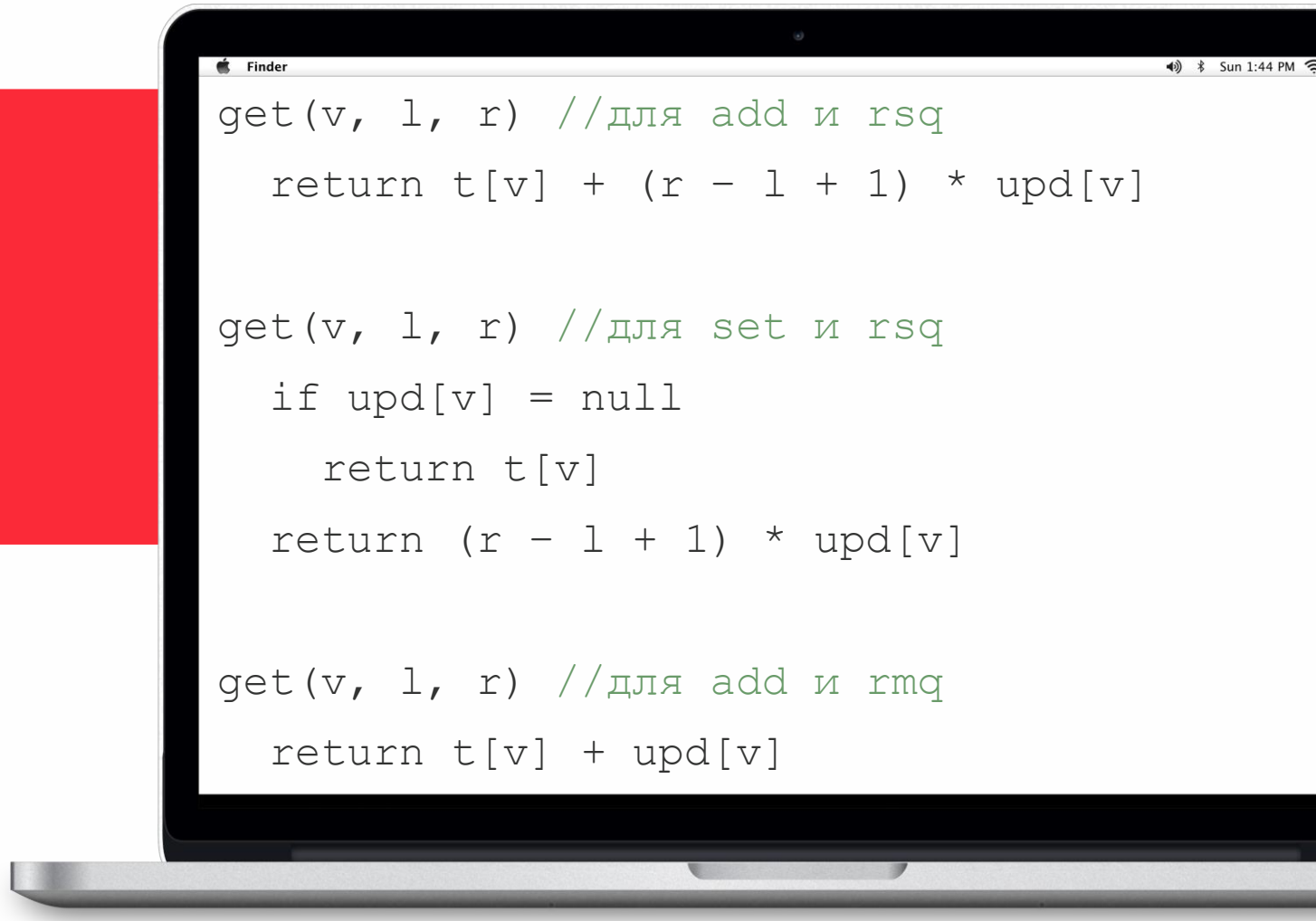
Дерево отрезков

- Групповое обновление
- Пусть [l, r] – отрезок, за который отвечает вершина v
- Три типа вершин
  - $[\mathrm{l}, \mathrm{r}] \in [a, b]$
  - $[\mathrm{l}, \mathrm{r}] \cap [a, b] = \emptyset$
  - $[\mathrm{l}, \mathrm{r}] \cap [a, b] \neq \emptyset$
- Добавим пометку в каждую вершину v – upd[v]



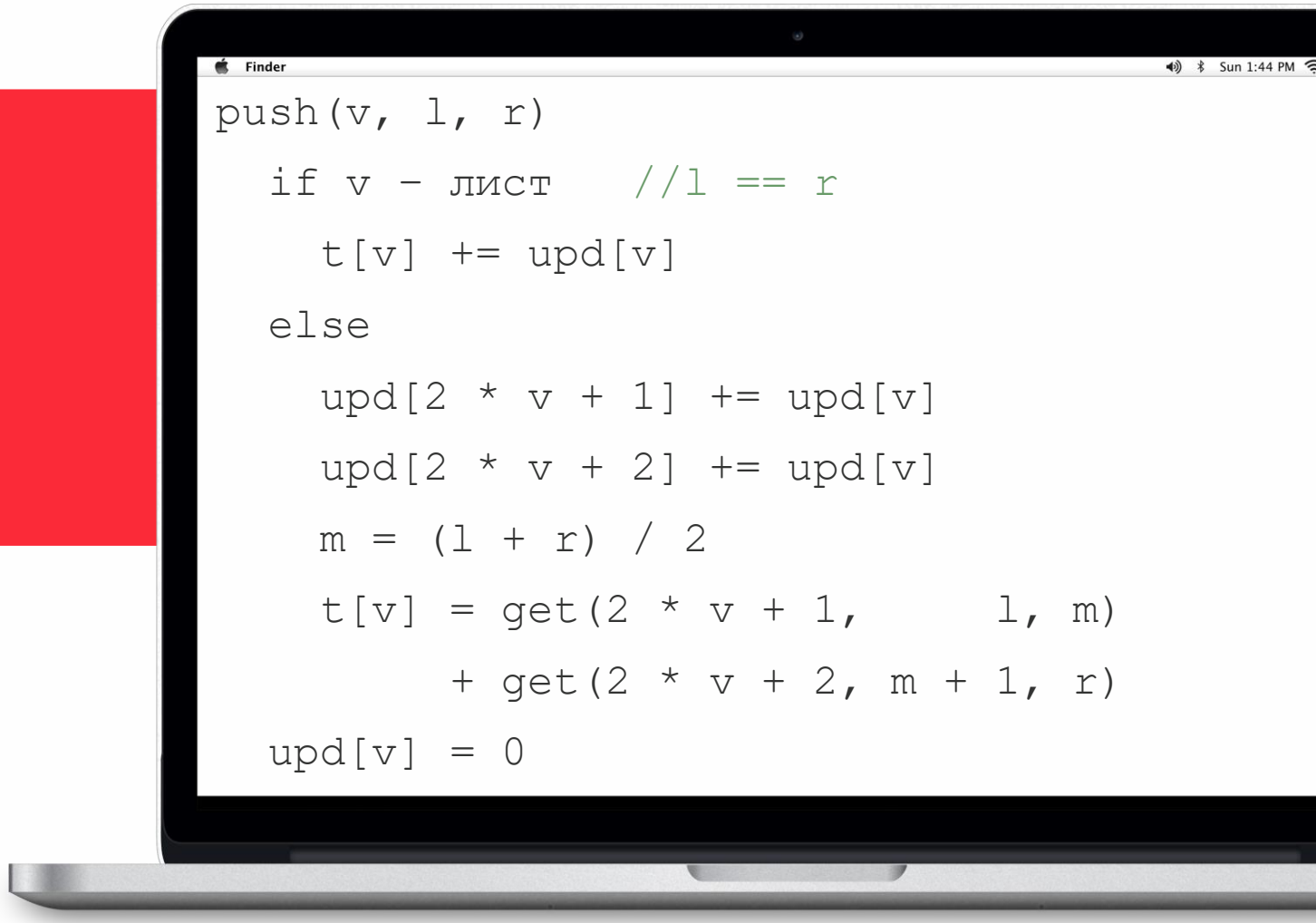
Дерево Отрезков

Групповое обновление:
вспомогательные функции



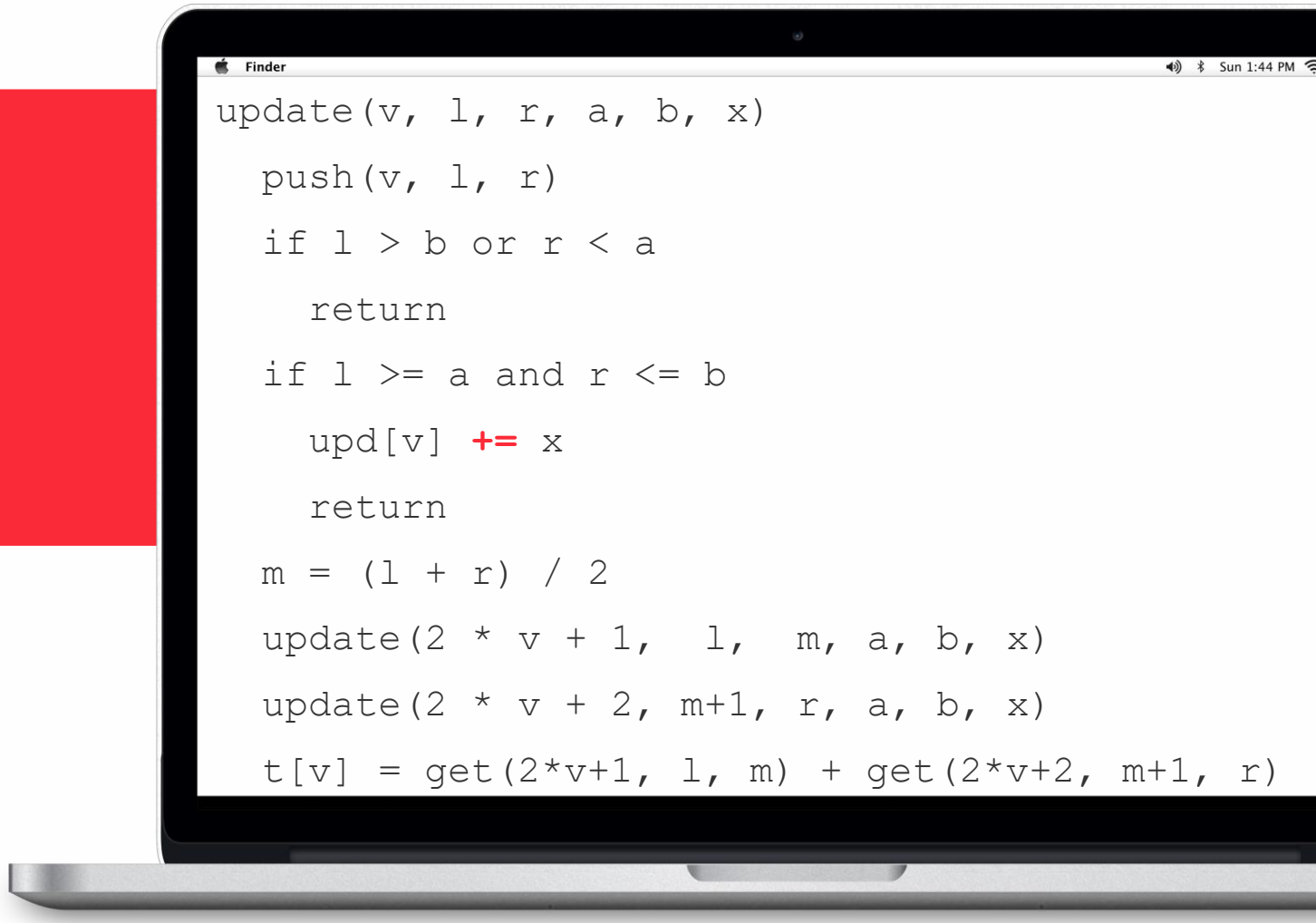
Дерево Отрезков

Групповое обновление:
вспомогательные функции



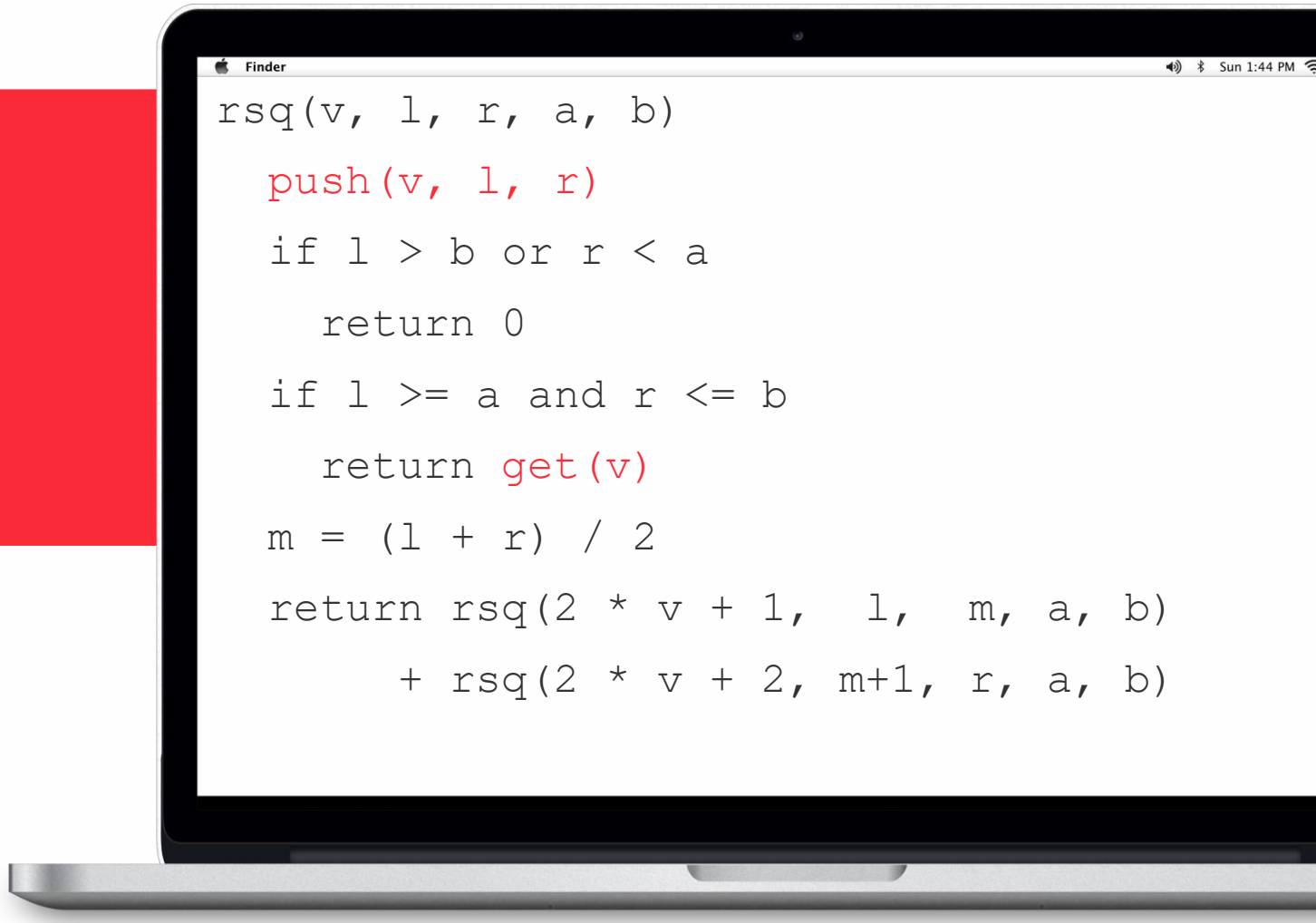
Дерево Отрезков

Групповое обновление



Дерево Отрезков

RSQ(a, b) при групповых обновлениях



Все!