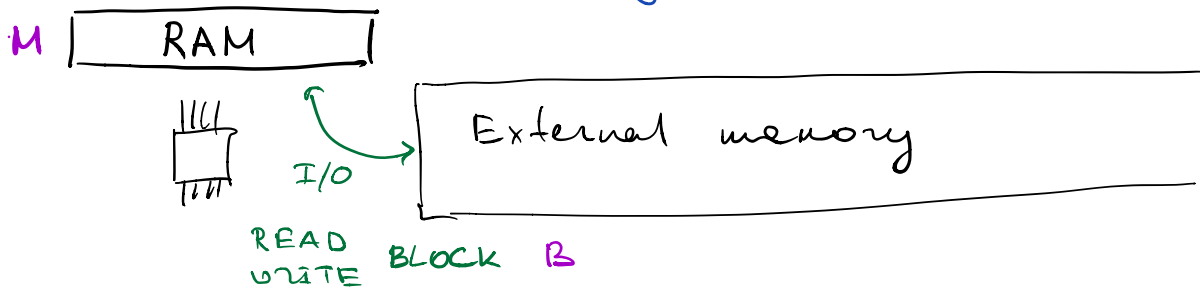


Альтернативные модели вычислений

Работа с большими данными

Модель внешней памяти
(External memory model)



Важнейший момент: чтение/запись происходит блоками

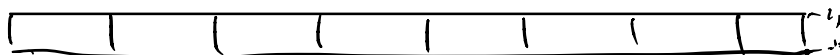
Сложность \equiv # I/O операций

Что значит эфф. алгоритм?

В RAM задача решается за $O(f(n))$
То же хотим её решить $O(f(n)/B)$

Почему э-та в массиве

B RAM за $O(n)$
B EM за $O(n/B)$

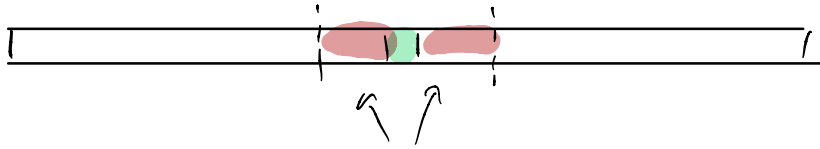


Форматный поиск в упоряд. массиве

$$n \rightarrow B \quad \log \frac{n}{B} = \log n - \log B$$

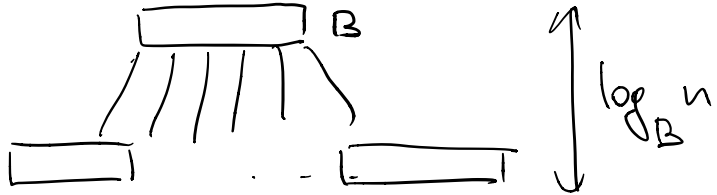
Сложность $O(\log_2 n)$

Хотелось бы $O(\log_B n)$



$B-1$ минимизировать n -ов

B -дерево

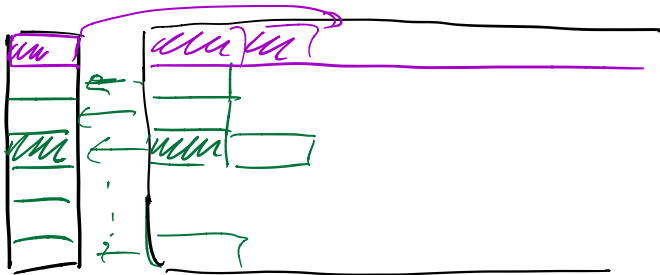


Сортировка слиянием



$$O\left(\frac{N}{B} \log N\right)$$

2-way merge sort



k-way merge sort

$$B \text{ RAM} \quad O(kN \log_k N)$$

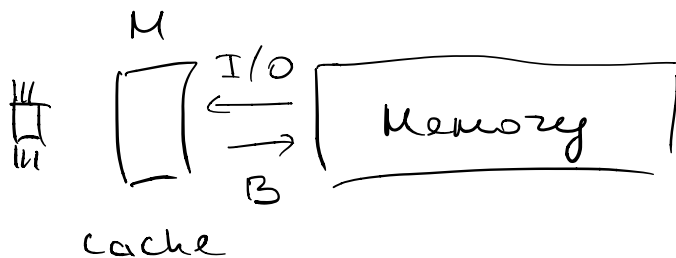
$$O(N \log_k N \cdot \log_2 k)$$

$$B \text{ EM} \quad O\left(\frac{N}{B} \log_k N\right)$$

$$k = \frac{M}{B}$$

$$B \text{ результаты: } O\left(\frac{N}{B} \cdot \log_{\frac{M}{B}} N\right)$$

Cache-oblivious



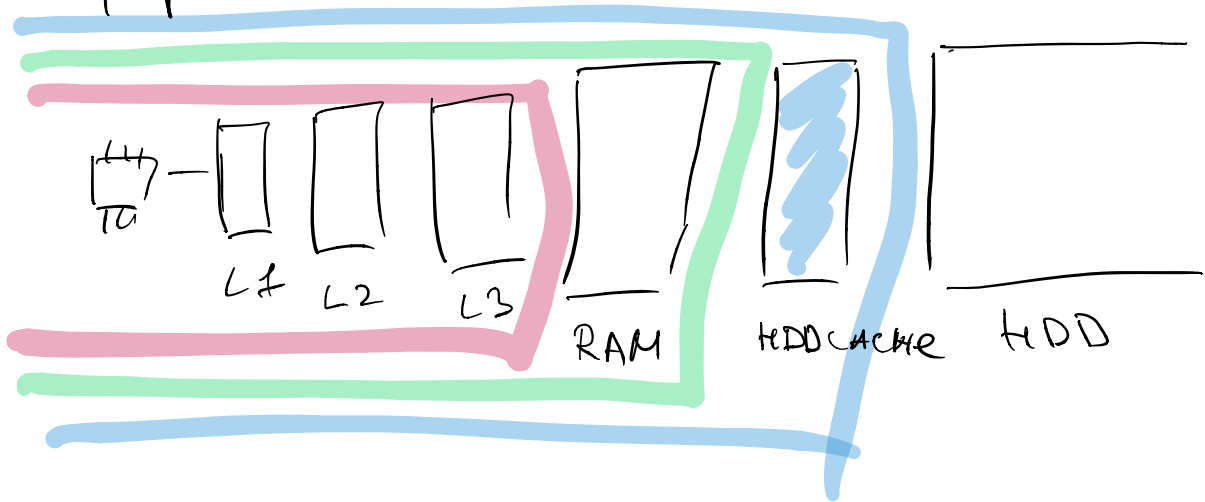
Отличие от EM

1. Мы не знаем M и B
2. Управление кэшем происходит незаметно для нас

(например, FIFO - стратегия,
LRU - стратегия)

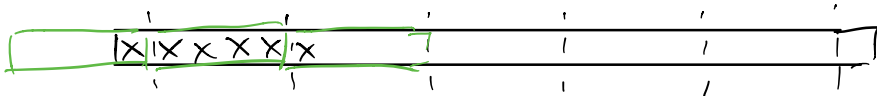
Сложность: $\#$ I/O операций

Если алгоритм эфф. в кэше-обл.
 памяти, то он эффект. на \forall
 иерархии кэшей



Почек в массиве

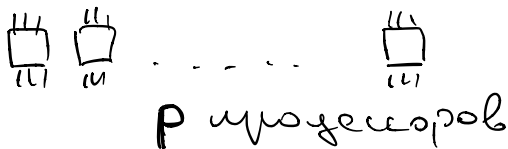
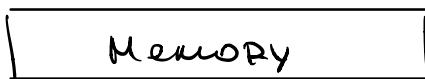
$$O(n/B)$$



$$\sim \lceil n/B \rceil + 1 = O(n/B)$$

Параметрические и распределённые
 вычисления

1. PRAM



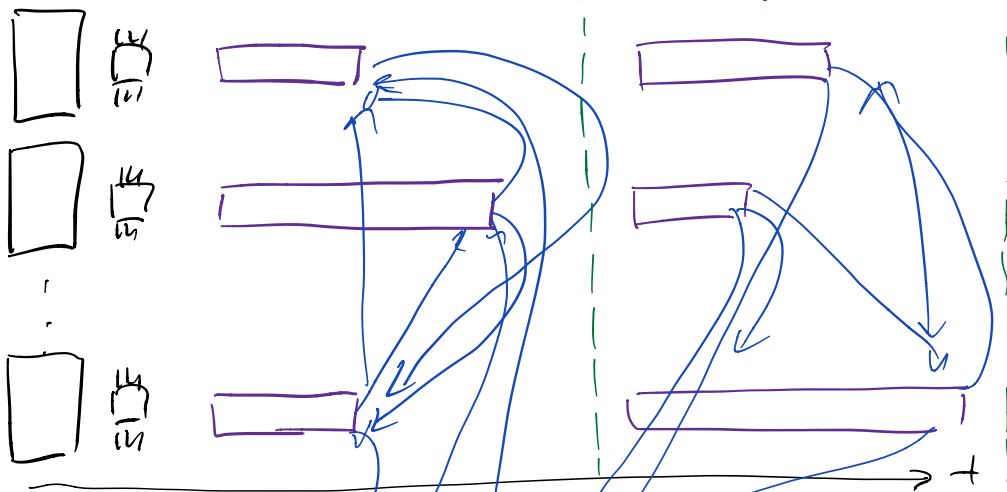
1. EREW
2. CREW
3. CRCW

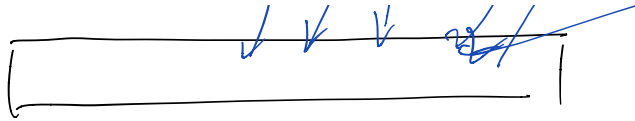
CRCW алгоритм дает точку max

$$p = n^2 \quad O(1)$$

1. for $i = 1$ to n : $O(1)$
maximum[i] = True
i -й элемент может быть max.
2. for $i = 1$ to n : $O(1)$
for $j = 1$ to n :
if $a[i] < a[j]$:
 - maximum[i] = False
3. for $i = 1$ to n : $O(1)$
if maximum[i]:
 - result = i

Модель BSP





Внешняя память

1. P процессоров (1 ед. времени на оп.)
2. коммуникационная среда
(G единиц времени на позицию
1 значение)
3. барьерный механизм синхрониз.
(L ед. времени на синхрониз.)

Суперкомпьютер Cray T3E

$$P = 64, \quad G \approx 78, \quad L \approx 1825$$

- $\text{comp}(S, P)$ — # операций процессора P на шаге S
- $\text{comm}(S, P)$ — # значений, посланных P на шаге S .

Стоимость одного шага?

- $\text{comp}(S) = \max_P \text{comp}(S, P)$
- $\text{comm}(S) = \max_P \text{comm}(S, P)$

- $\text{cost}(s) = \text{comp}(s) + G \cdot \text{comm}(s) + L$
- $\text{cost} = \sum_s \text{cost}(s)$

Что нам хотим?

Если в RAM $O(f(n))$ и $O(g(n))$
то в BSP: I/O опер.

$$\text{comp} = O(f(n)/p)$$

$$\text{comm} = O(g(n)/p)$$

$$\text{sync} = O(h(p))$$

$$\text{sync} = O(1) \quad \leftarrow \text{не зависит от } n$$

Поиск x -го в массиве

1. x процессор считывает $\frac{n}{p}$ элементов

$$\text{comm}(1) = O\left(\frac{n}{p}\right)$$

$$\text{comp}(1) = O(1)$$

2. Все сообщения идут
min вхождение в выд. процессору

$$\text{comp}(2) = O\left(\frac{n}{p}\right)$$

$$\text{comm}(2) = O(p)$$

3. Выд. процессор выдает

так и у полученных корней
и дробных. это в первую
очередь

$$\text{comp}(z) = O(p)$$

$$\text{comm}(z) = O(1)$$

$$\text{comp} = O(u/p)$$

$$\text{comm} = O(u/p)$$

$$\text{sync} = O(1)$$