

Алгоритмы для работы с большими объемами данных, теория 1

Задача 1 (1 балл)

Заданы числа a_1, \dots, a_N , записанные на диск. В дополнение к записанным числам необходимо построить структуру данных размера $O(N/B)$, которая бы позволяла в реальном времени отвечать на запросы поиска **суммы** на отрезке $[i, j]$. Для ответа на запрос должно использоваться константное количество обращений к диску. За какое время (количество обращений к диску) такая структура данных может быть построена?

Задача 2 (3 балла)

Заданы числа a_1, \dots, a_N , записанные на диск, также известно, что $\log_2 N < B$. В дополнение к записанным числам необходимо построить структуру данных размера $O(N/B)$, которая бы позволяла в реальном времени отвечать на запросы поиска **минимума** на отрезке $[i, j]$. Для ответа на запрос должно использоваться константное количество обращений к диску. За какое время (количество обращений к диску) такая структура данных может быть построена?

Задача 3 (3 балла)

Заданы числа a_1, \dots, a_N , записанные на диск, а также задана последовательность из M запросов. Запросы бывают двух видов:

1. увеличить все числа на отрезке $[i, j]$ на величину c ;
2. найти минимум на отрезке $[i, j]$.

Требуется построить алгоритм, который оффлайн обработает все запросы в заданном порядке и запишет ответы (на вопросы второго типа) на диск. Время работы алгоритма не должно превосходить $O(\text{Sort}((N + M) \log_2 N))$. Известно, что $\log_2 N$ не превосходит размера оперативной памяти.

Задача 4 (2 балла)

Задан взвешенный граф $G = (V, E)$, записанный на диск, а также известно, что $|V| < M$. Постройте алгоритм, использующий не более $O(\text{Scan}(|E| + |V|))$ обращений к диску, который найдет минимальное остовное дерево в заданном графе. Какое время работы будет у данного алгоритма с точки зрения количества операций процессора? Покажите, как получить время работы равное $O(|E| \log |V|)$.

Задача 5 (1 балл)

На диске записан оргграф $G = (V, E)$, представляющий собой множество непересекающихся ориентированных деревьев. Постройте алгоритм, использующий не более $O(\text{Sort}(|V|))$ обращений к диску, который для каждой вершины графа найдет номер корня, соответствующего ей дерева.