Algorithms. SET 5. A4

Algorithms SET 5

`A --> F(A)`
`B --> F(B)`

`F(A) & F(B) --> F(AB)`

Фильтр Блума при добавлении элемента ставит в битовом векторе `1`. В местах, куда никакой элемент не попал, стоит `0`.

Вопрос 1: `A ∩ B ∈ F(AB)`?

Фильтр Блума `F(AB)` после применения побитового И будет иметь `1`, там, где совпали биты `F(A)` и `F(B)`, то есть те биты, которые были и в `F(A)`, и `F(B)`.

Рассмотрим множество `A ∩ B`. Оно состоит только из элементов, которые есть и в множестве `A`, и в множестве `B`. Тогда их биты точно есть и в `F(A)`, и `F(B)`. А так как они есть в обоих фильтрах, то они также будут и в фильтре `F(AB)`.

Следовательно, `F(AB)` всегда **будет выдавать положительные ответы** о принадлежности объектов из множества `A ∩ B`.

Ответ: да.

Вопрос 2: `F(A ∩ B) == F(AB)`?

Рассмотрим множество `A = {a, b}` и множество `B = {a, c}`.
Пусть:

- `f1(a) = 0`, `f2(a) = 2`
- `f1(b) = 1`, `f2(b) = 2`
- `f1(c) = 3`, `f2(c) = 1`

Тогда:
`F(A) = F({a, b}) = [1, 1, 1, 0]`.
`F(B) = F({a, c}) = [1, 1, 1, 1]`.

Побитовое И: `F(AB) = F(A) & F(B) = [1, 1, 1, 0]`.

Пересечение множеств: `A & B = {a}`.
Тогда `F(A & B) = F({a}) = [1, 0, 1, 0]`.

Но `[1, 1, 1, 0] ≠ [1, 0, 1, 0] => F(AB) ≠ F(A & B)` - противоречие.

Фильтр в точности не соответствует, так как при применении побитового И в фильтре могут появиться лишние биты, которые получились из одной хеш-функции для элементов множества `A` и из другой хеш-функции для элементов множества `B`, но дающих одинаковый ответ. Это приведёт к появлению ложно-положительных ответов.

Ответ: нет.