

Динамика по подмножествам

Иван Сергеевич Казменко

Санкт-Петербургский Государственный Университет

Пятница, 29 марта 2024 года

Содержание

- 1 Отметки на подмножествах
 - Постановка задачи
 - Пример
 - Решение
- 2 Количество отметок
 - Постановка задачи
 - Пример
 - Решение
- 3 Векторизация битовых операций
 - Постановка задачи
 - Решение

Содержание

- 1 Отметки на подмножествах
 - Постановка задачи
 - Пример
 - Решение
- 2 Количество отметок
 - Постановка задачи
 - Пример
 - Решение
- 3 Векторизация битовых операций
 - Постановка задачи
 - Решение

Постановка задачи

Постановка задачи:

- Рассмотрим все подмножества множества из n элементов.
- Отметим некоторые подмножества.
- Распространим отметку с каждого подмножества на все его надмножества.
- Какие подмножества теперь отмечены?

Постановка задачи

Постановка задачи:

- Рассмотрим все подмножества множества из n элементов.
- Отметим некоторые подмножества.
- Распространим отметку с каждого подмножества на все его надмножества.
- Какие подмножества теперь отмечены?

Обозначения:

- Пронумеруем элементы множества числами $0, 1, \dots, n - 1$.
- Будем записывать подмножества как строки из n двоичных цифр.
- Цифра 0 в i -м разряде означает отсутствие элемента i .
- Цифра 1 в i -м разряде означает присутствие элемента i .
- Например, при $n = 5$ множество $\{1, 3, 4\}$ будем записывать как 11010.

Пример

Элементы
с единицы:

4 5
2 1 2
3 1 2 3
2 1 4
3 1 3 4
2 2 4

Двоичные
строки:

4 5
0011
0111
1001
1101
1010

0000: x x
0001: x x
0010: x x
0011: v v
0100: x x
0101: x x
0110: x x
0111: v v
1000: x x
1001: v v
1010: v v
1011: x v
1100: x x
1101: v v
1110: x v
1111: x v

Решение

Чтение входных данных:

```
1      int n, x;
2      cin >> n >> x;
3      int m = 1 << n;
4
5      bool mark [m];
6      fill (mark, mark + m, false);
7      for (int i = 0; i < x; i++) {
8          int s = 0, t, v;
9          cin >> t;
10         for (int j = 0; j < t; j++) {
11             cin >> v;
12             s |= 1 << (v - 1);
13         }
14         mark[s] = true;
15     }
```

Решение

Наивное решение за $O(4^n)$:

```
1     for (int s = 0; s < m; s++)
2         if (mark[s])
3             for (int t = 0; t < m; t++)
4                 if ((t & s) == s)
5                     mark[t] = true;
```

Решение

Можно поменять порядок циклов:

```
1     for (int t = 0; t < m; t++)
2         for (int s = 0; s < m; s++)
3             if ((t & s) == s)
4                 if (mark[s])
5                     mark[t] = true;
```

Решение

Перебор подмножеств подмножеств за $O(3^n)$:

```
1     for (int t = 0; t < m; t++) {
2         int s = t;
3         do {
4             if (mark[s])
5                 mark[t] = true;
6                 s = (s - 1) & t;
7         } while (s != t);
8     }
```

Решение

Короче, но без пустого множества:

```
1     for (int t = 0; t < m; t++)
2         for (int s = t; s > 0; s = (s - 1) & t)
3             if (mark[s])
4                 mark[t] = true;
```

Решение

Динамическое программирование за $O(2^n \cdot n)$:

```
1     for (int s = 0; s < m; s++)
2         for (int k = 0; k < n; k++)
3             if (!(s & (1 << k)))
4                 mark[s | (1 << k)] |= mark[s];
```

Решение

Можно считать с другой стороны:

```
1     for (int t = 0; t < m; t++)
2         for (int k = 0; k < n; k++)
3             if (t & (1 << k))
4                 mark[t] |= mark[t ^ (1 << k)];
```

Решение

Можно поменять порядок циклов:

```
1     for (int k = 0; k < n; k++)
2         for (int t = 0; t < m; t++)
3             if (t & (1 << k))
4                 mark[t] |= mark[t - (1 << k)];
```

Содержание

- 1 Отметки на подмножествах
 - Постановка задачи
 - Пример
 - Решение
- 2 Количество отметок
 - Постановка задачи
 - Пример
 - Решение
- 3 Векторизация битовых операций
 - Постановка задачи
 - Решение

Постановка задачи

Постановка задачи:

- Рассмотрим все подмножества множества из n элементов.
- Отметим некоторые подмножества.
- Для каждого подмножества нужно вычислить, сколько его подмножеств отмечены.

Пример

Элементы с единицы:	Двоичные строки:	
		0000: 0
		0001: 0
		0010: 0
4 5	4 5	0011: 1
2 1 2	0011	0100: 0
3 1 2 3	0111	0101: 0
2 1 4	1001	0110: 0
3 1 3 4	1101	0111: 2
2 2 4	1010	1000: 0
		1001: 1
		1010: 1
		1011: 3
		1100: 0
		1101: 2
		1110: 1
		1111: 5

Решение

Чтение входных данных:

```
1      int n, x;
2      cin >> n >> x;
3      int m = 1 << n;
4
5      int num [m];
6      fill (num, num + m, 0);
7      for (int i = 0; i < x; i++) {
8          int s = 0, t, v;
9          cin >> t;
10         for (int j = 0; j < t; j++) {
11             cin >> v;
12             s |= 1 << (v - 1);
13         }
14         num[s] += 1;
15     }
```

Решение

Неправильное решение – почему?

```
1     for (int s = 0; s < m; s++)
2         for (int k = 0; k < n; k++)
3             if (!(s & (1 << k)))
4                 num[s | (1 << k)] += num[s];
```

Решение

Правильное решение за $O(2^n \cdot n)$:

```
1     for (int k = 0; k < n; k++)
2         for (int s = 0; s < m; s++)
3             if (!(s & (1 << k)))
4                 num[s | (1 << k)] += num[s];
```

Содержание

- 1 Отметки на подмножествах
 - Постановка задачи
 - Пример
 - Решение
- 2 Количество отметок
 - Постановка задачи
 - Пример
 - Решение
- 3 Векторизация битовых операций
 - Постановка задачи
 - Решение

Постановка задачи

- Задача – распространить отметки по подмножествам.
- Будем хранить каждую отметку в одном бите.

Решение

Чтение входных данных:

```
1      int n, x;
2      cin >> n >> x;
3      int upperN = max (n - 5, 0);
4      int m = (1 << upperN);
5
6      vector <int> mark (m);
7      for (int i = 0; i < x; i++) {
8          int s = 0, t, v;
9          cin >> t;
10         for (int j = 0; j < t; j++) {
11             cin >> v;
12             s |= 1 << (v - 1);
13         }
14         mark[s >> 5] |= 1 << (s & 31);
15     }
```

Решение

Решение за $O((2^n \cdot n)/32)$:

```
1      for (int s = 0; s < m; s++) {
2          mark[s] |= (mark[s] & 0x55555555) << 1;
3          mark[s] |= (mark[s] & 0x33333333) << 2;
4          mark[s] |= (mark[s] & 0x0F0F0F0F) << 4;
5          mark[s] |= (mark[s] & 0x00FF00FF) << 8;
6          mark[s] |= (mark[s] & 0x0000FFFF) << 16;
7          for (int k = 0; k < upperN; k++)
8              if (!(s & (1 << k)))
9                  mark[s | (1 << k)] |= mark[s];
10     }
```

Решение

Пояснение:

```
2          mark[s] |= (mark[s] & 0x55555555) << 1;
```

```
w = 0x55555555 = *.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*
y = x & w      = .u.s.q.o.m.k.i.g.e.c.a.8.6.4.2.0
x              = vutsrqponmlkjihgfedcba9876543210
z = y << 1     = u.s.q.o.m.k.i.g.e.c.a.8.6.4.2.0.
```

Решение

Пояснение:

```
3      mark[s] |= (mark[s] & 0x33333333) << 2;
```

```
w = 0x33333333 = ..**..**..**..**..**..**..**..**..**
y = x & w      = ..ts..po..lk..hg..dc..98..54..10
x              = vutsrqponmlkjihgfedcba9876543210
z = y << 2     = ts..po..lk..hg..dc..98..54..10..
```

Решение

Пояснение:

```
4          mark[s] |= (mark[s] & 0x0F0F0F0F) << 4;
```

```
w = 0x0F0F0F0F = ....****....****....****....****
```

```
y = x & w      = ....rqpo....jihg....ba98....3210
```

```
x              = vutsrqponmlkjihgfedcba9876543210
```

```
z = y << 4     = rqpo....jihg....ba98....3210....
```

Решение

Пояснение:

```
5      mark[s] |= (mark[s] & 0x00FF00FF) << 8;
```

```
w = 0x00FF00FF = .....*****.....*****
y = x & w      = .....nmlkjihg.....76543210
x              = vutsrqponmlkjihgfedcba9876543210
z = y << 8     = nmlkjihg.....76543210.....
```

Решение

Пояснение:

```
6      mark[s] |= (mark[s] & 0x0000FFFF) << 16;
```

```
w = 0x0000FFFF = .....*****
y = x & w      = .....fedcba9876543210
x              = vutsrqponmlkjihgfedcba9876543210
z = y << 16    = fedcba9876543210.....
```

Вопросы?

Вопросы?