acm.mipt.ru

олимпиады по программированию на Физтехе

Раздел «Алгоритмы» . FenwickTreeCPP :

Дерево Фенвика и Двоичный контейнер

- Файл macros.h с макросами STL

Дерево Фенвика предоставляет возможность хранить массив A[0]... A[N-1] и извлекать/менять его содержимое по следующему интерфейсу:

- sum(i) – сумма элементов A[0] + A[1] + ... +A[i]
- change(i, v) – выполнить действие A[i] += v

Ниже представлен код, с реализацией этого интерфейса

- dummy – простейшая реализация интерфейса "в лоб"
- ftree – дерево Фенвика
- bctree – двоичный контейнер

```
#include "macros.h"

struct dummy {
        vi A;
        dummy(int N) : A(vi(N, 0)) {
        }
        void change(int i, int v) {
                A[i] += v;
        }
        int sum(int i) {
                return accumulate(A.begin(), A.begin() + i + 1, 0);
        }
};

struct ftree {
        vi B;
        ftree(int N) : B(vi(N, 0)) {
        }
        void change(int i, int v) {
                B[i] += v;
                int mask = 1;
                while(true) {
                        if(!(i & mask)) {
                                i |= mask;
                                if(i >= sz(B)) {
                                        break;
                                }
                                B[i] += v;
                        }
                        mask <<= 1;
                }
        }
        int sum(int i) {
                int r = 0;
                while(i >= 0) {
                        r += B[i];
                        i = (i&(i+1))-1;
                }
                return r;
        }
};

struct ftree2 {
        vi B;
        ftree2(int N) : B(vi(N, 0)) { }
        void change(int i, int v) { while(B[i] += v, i |= i+1, i < sz(B)); }
        int sum(int i) { int r = 0; while(i >= 0) r += B[i], i = (i&(i+1))-1; return r; }
};
```

Поиск
Поиск
Раздел «Алгоритмы»
Главная
Форум
Ссылки
EI Judge
Инструменты:
Поиск
Изменения
Index
Статистика
Разделы
Информация
Алгоритмы
Язык Си
Язык Ruby
Язык Ассемблера
EI Judge
Парадигмы
Образование
Сети
Objective C
Logon>>

```
struct bctree {
        vi C;
        int O;
        bctree(int _N) {
                int N = 1;
                while(N < _N) N *= 2;
                C = vi(N*2);
                O = N;
        }

        void rec_change(int i, int v) {
                if(i > 0) {
                        C[i] += v;
                        rec_change(i/2, v);
                }
        }
        void change(int i, int v) {
                rec_change(O+i, v);
        }

        int rec_sum(int i) {
                if(i > 1) {
                        return (i%2 ? C[i-1] : 0) + rec_sum(i/2);
                }
                else {
                        return 0;
                }
        }

        int sum(int i) {
                return C[O+i] + rec_sum(O+i);
        }
};

struct bctree2 {
        vi C;
        int O;
        bctree2(int _N) {
                int N = 1;
                while(N < _N) N *= 2;
                C = vi(N*2);
                O = N;
        }

        void change(int i, int v) {
                i += O;
                while(i > 0) {
                        C[i] += v;
                        i /= 2;
                }
        }

        int sum(int i) {

                i += O;

                int res = C[i];

                while(i > 1) {
                        if(i%2) res += C[i-1];
                        i /= 2;
                }

                return res;
        }
};

struct bctree3 {
        vi C;
        int O;
        bctree3(int _N) {
                int N = 1;
```

```
                while(N < _N) N *= 2;
                C = vi(N*2);
                O = N;
        }
        void change(int i, int v) {
                i += O;
                while(C[i] += v, i /= 2);
        }
        int sum(int i) {
                i += O;
                int res = C[i];
                while((i%2 ? res += C[i-1] : 0), i /= 2);
                return res;
        }
};

struct bctree4 {
        vi C;
        int O;
        bctree4(int _N) {
                int N = _N;
                // These two lines may not be commented!
                //int N = 1;
                //while(N < _N) N *= 2;
                C = vi(N*2);
                O = N;
        }
        void change(int i, int v) {
                i += O;
                while(C[i] += v, i /= 2);
        }
        int sum(int i) {
                i += O;
                int res = C[i];
                while((i%2 ? res += C[i-1] : 0), i /= 2);
                return res;
        }
};

template<typename T1, typename T2> void test_ftree(int N = 1000) {
        T1 o1(N);
        T2 o2(N);
        loop(t, 1000) {
                int i = rand() % N;
                int v = (rand() % 201) - 100;
                o1.change(i, v);
                o2.change(i, v);
                int k = rand()%N;
                if(o1.sum(k) != o2.sum(k)) {
                        L("Test failed!\n");
                }
        }
        L("Test passed.\n");
}

int main() {
        L("ftree test\n");
        //test_ftree<dummy, ftree>();
        //test_ftree<ftree, ftree2>();
        //test_ftree<dummy, ftree2>();
        test_ftree<ftree2, bctree>();
        test_ftree<ftree2, bctree2>();
        test_ftree<ftree, bctree3>();
        test_ftree<ftree, bctree4>(1024);
}
```