

Раздел «Алгоритмы» . GenerateCPP :

Переборные алгоритмы $n!$, 2^n , $C(n,m)$ на C

- теория
- на Pascal

Содержание

- Переборные алгоритмы $n!$, 2^n , $C(n,m)$ на C
 - Перебор 2^n — все подмножества
 - Перебор C_n^m — все k -элементные подмножества
 - Перебор $n!$

Перебор 2^n — все подмножества

Дано n -элементное множество. Нужно перебрать все подмножества. Количество подмножеств n -элементного подмножества равно 2^n .

```
M = intpower(2,n);
for(i=0; i < M; i++)
{
    int a = i;
    // a — зашифрованное подмножество
    // j-й бит числа a соответствует j-у элементу
    for(j=0; j < n ; j++)
    {
        if(a%2){
            // берем элемент с номером j
        }
        a /=2;
    }
}
```

В более общем случае приходится использовать массив A, в котором раныть нули и единицы.

- $A[i] = 1$ означает, что элемент с номером i "взят"
- $A[i] = 0$ означает, что элемент с номером i не "взят"

Удобнее всего для перебора использовать рекурсивные функции.

```
int A[M] = {0,..};
void gen_2n(int i)
{
    if(i==n) {
        // рассматриваем подмножество, задаваемое массивом A
    }
    A[i] = 1; gen_2n(i+1);
    A[i] = 0; gen_2n(i+1);
}
```

Перебор C_n^m — все k -элементные подмножества

Для решения данной задачи подмножества удобнее представлять в виде упорядоченного набора из k номеров элементов.

```
// Максимальное значение m
#define M 100
```

Поиск

Поиск

Раздел
«Алгоритмы»

Главная
Форум
Ссылки
El Judge

Инструменты:

Поиск
Изменения
Index
Статистика

Разделы

Информация
Алгоритмы
Язык Си
Язык Ruby
Язык
Ассемблера
El Judge
Парадигмы
Образование
Сети
Objective C

Login>>

```
int m,n;
int A[M+1] = {0,..};

void genCnm(int i)
{
    if(i > m) {
        // рассматриваем подмножество, задаваемое массивом A
    }
    for(j=A[i-1]; j < n ; j++)
    {
        A[i] = m; genCnm(i+1);
    }
}
main()
{
    scanf("%d%d", &k, &n);
    gen_Cnm(1);
}
```

Перебор $n!$

```
int A[M] = {0,..};
int n;
void gen_Fact_n(int i)
{
    if(i==n) {
        // рассматриваем подмножество, задаваемое массивом A
    }
    for(j=0; j < n ; j++)
        if(!was[j])
        {
            was[j] = 1;
            A[i] = j; gen_Fact_n(i+1);
            was[j] = 0;
        }
}
main()
{
    scanf("%d", &n);
    gen_Fact_n(0);
}
```

Copyright © 2003-2022 by the contributing authors.