NCKU Programming Contest Training Course
Course 6
2013/01/24

Pin-Chieh Huang (free999)

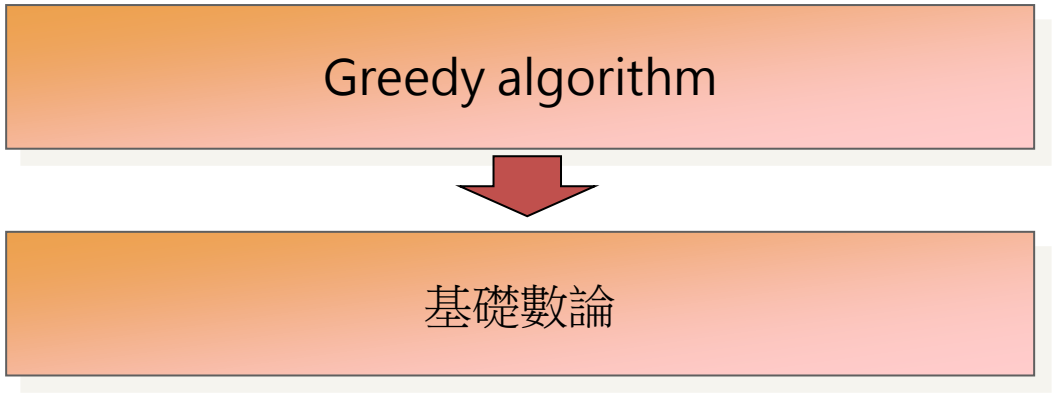
Pinchieh.huang@gmail.com

http://myweb.ncku.edu.tw/~p76014143/Course6.rar

Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan

made by electron & free999 & kk & rabbit125

Outline



made by electron & free999 & kk & rabbit125

Greedy Strategy

- Greedy Strategy
  - Choose an Optimal Solution in Now-Running Step
  - **Greedy algorithms are simple and straightforward**. They are shortsighted in their approach in the sense that they take decisions on the basis of information at hand without worrying about the effect these decisions may have in the future. They are easy to invent, easy to implement and most of the time quite efficient. Many problems cannot be solved correctly by greedy approach. Greedy algorithms are used to solve optimization problems

Greedy Strategy

- Example
  - Minimum Number of Changing Problem
    - Make a change of a given amount using the smallest possible number of coins.
  - Available Coins
    - dollars (100 cents)
    - quarters (25 cents)
    - dimes (10 cents)
    - nickels (5 cents)
    - pennies (1 cent)
  - **Make a change of a given amount using the smallest possible number of coins**

made by electron & free999 & kk & rabbit125

Example - 1

POJ 3617 Best Cow Line

Problem Description

FJ is about to take his N ($1 \le N \le 2{,}000$) cows to the annual"Farmer of the Year" competition. In this contest every farmer arranges his cows in a line and herds them past the judges.

The contest organizers adopted a new registration scheme this year: simply register the initial letter of every cow in the order they will appear (i.e., If FJ takes Bessie, Sylvia, and Dora in that order he just registers BSD). After the registration phase ends, every group is judged in increasing lexicographic order according to the string of the initials of the cows' names.

FJ is very busy this year and has to hurry back to his farm, so he wants to be judged as early as possible. He decides to rearrange his cows, who have already lined up, before registering them.

FJ marks a location for a new line of the competing cows. He then proceeds to marshal the cows from the old line to the new one by repeatedly sending either the first or last cow in the (remainder of the) original line to the end of the new line. When he's finished, FJ takes his cows for registration in this new order.

Given the initial order of his cows, determine the least lexicographic string of initials he can make this way.

Example - 1

POJ 3617 Best Cow Line

IO Description

The number of participants, n: 3 ≤ n ≤ 10000.
The distance (measured in centimeters), d: 500 ≤ d ≤ 200000.
The running speed (centimeters per second) of each participant, ri: 50 ≤ ri ≤ 1000.

Input

Line 1: A single integer: *N*
Lines 2..*N*+1: Line *i*+1 contains a single initial ('A'..'Z') of the cow in the *i*th position in the original line

Output

The least lexicographic string he can make. Every line (except perhaps the last one) contains the initials of 80 cows ('A'..'Z') in the new line.

made by electron & free999 & kk & rabbit125

Example - 1

POJ 3617 Best Cow Line

Sample I/O

Sample Input

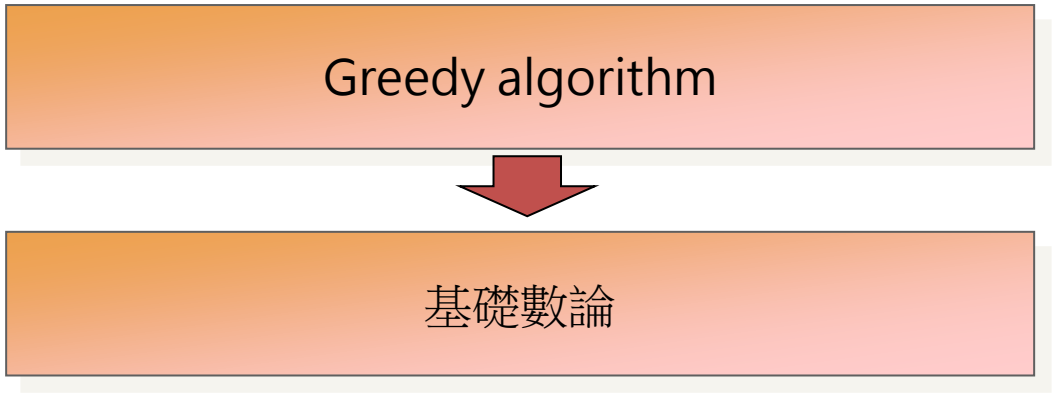
```
6
A
C
D
B
C
B
```

Sample Output

```
ABCBCD
```


made by electron & free999 & kk & rabbit125

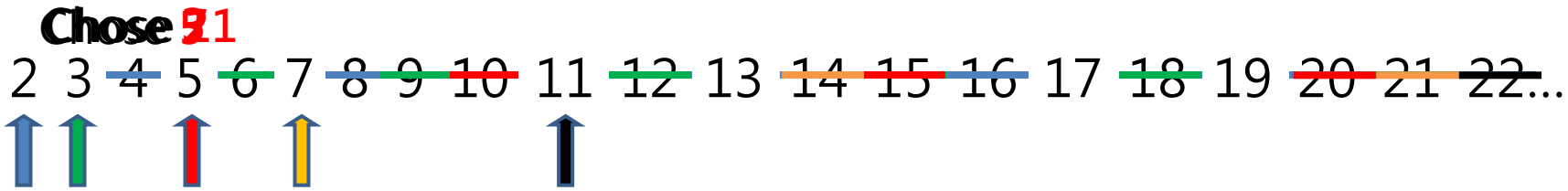
Outline



made by electron & free999 & kk & rabbit125

Prime Number

- We use **sieve** to create a prime array
  - Chose the smallest number at each iteration and delete the multiple of this number




```
#define MAX 1000000
bool isprime[MAX];
void Sieve(){                                          // Time complexity may be (n x sqrt(n))
        memset( isprime, true, sizeof(isprime) );
        isprime[0] = false;
        isprime[1] = false;
        for( int i=2; i<MAX; i++)          ⟹  for( int i=2; i<=sqrt(MAX); i++)
                if( isprime[i] )
                        for(int j = i + i; j<MAX; j += i)   ⟹  for( int j=i * i; j<MAX; j += i )
                                isprime[j] = false;
}
```


made by electron & free999 & kk & rabbit125

Prime Number

- For advanced:
- http://www.csie.ntnu.edu.tw/~u91029/Prime.html

- But just remember that the code in previous page is fast enough to solve almost every prime problems

Euler's Phi Function

- Also called Euler's totient *function*
- The **totient** of a positive integer n is defined to be
  - the number of positive integers <= n that are coprime to n.
  - For example, 1, 2, 4, 5, 7 and 8 are coprime to 9.

- To find totient of 12, we have to factorize 12 first.
  - 12 = 2^2 * 3^1
  - How to factorize it?
  - How to count the number of factor of 12
  - How to count the phi(12)

made by electron & free999 & kk & rabbit125

Euler's Phi Function

- Test n with every primes
- If prime i can be divided by n, count the power

```
int n = 12, pos = 0;
int tmp = n;
for( int i = 2;i <= (int)sqrt(n); i++){
        if(!isprime[i] || tmp % i != 0)continue;        // if I is not prime or tmp cannot divided by i
        p[pos] = i;                                     // p[pos] = prime i
        while(tmp % i == 0) tmp /= i, a[pos]++;         // increase the power a[pos]
        pos++;
}
printf("%d =", n);
for( int i = 0;i < pos; i++){
        if(i) printf(" x");
        printf(" %d ^ %d", p[i], a[i]);                 // n = p0^a0 + p1^a1 + ... + p(pos - 1) ^ a(pos - 1)
}
```

```
12 = 2 ^ 2 x 3 ^ 1                                      // stdout
```


made by electron & free999 & kk & rabbit125

Example 2

NCKU Judge 21 Goldbach's Conjecture

Description:

In 1742, Christian Goldbach, a German amateur mathematician, sent a letter to Leonhard Euler in which he made the following conjecture:
Every even number greater than 4 can be
written as the sum of two odd prime numbers.
For example:
8 = 3 + 5. Both 3 and 5 are odd prime numbers.
20 = 3 + 17 = 7 + 13.
42 = 5 + 37 = 11 + 31 = 13 + 29 = 19 + 23.
Today it is still unproven whether the conjecture is right. (Oh wait, I have the proof of course, but it is too long to write it on the margin of this page.)
Anyway, your task is now to verify Goldbach's conjecture for all even numbers less than a million.

made by electron & free999 & kk & rabbit125

Example 2

NCKU Judge 21 Goldbach's Conjecture

Input:
The input will contain one or more test cases.
Each test case consists of one even integer n with 6 <= n < 1000000.
Input will be terminated by a value of 0 for n.

Output:
For each test case, print one line of the form n = a + b, where a and b are odd primes. Numbers and operators should be separated by exactly one blank like in the sample output below. If there is more than one pair of odd primes adding up to n, choose the pair where the difference b - a is maximized. If there is no such pair, print a line saying "Goldbach's conjecture is wrong."

made by electron & free999 & kk & rabbit125

Example 2

NCKU Judge 21

Sample test:

input

```
8
20
42
0
```

output

```
8 = 3 + 5
20 = 3 + 17
42 = 5 + 37
```


made by electron & free999 & kk & rabbit125

Homework 6

Total 36 problem

- **Greedy algorithm**

  Uva

  120, 311, 10249, 10037, 11269, 11729

- **Prime**

  NCKU Judge

  20, 21, 23, 24, 26, 27, 72, 183, 193, 210, 231

  Uva

  1210, 11064, 10311, 10622, 10290, 10791, 10780, 583, 10299, 10140, 524, 160, 543, 10539, 406, 10924

  POJ

  2262, 2739, 3006

made by electron & free999 & kk & rabbit125

Thank for Your Attention

made by electron & free999 & kk & rabbit125