Thuật toán ứng dụng

TƯ DUY THUẬT TOÁN VÀ CTDL + KỸ NĂNG LẬP TRÌNH

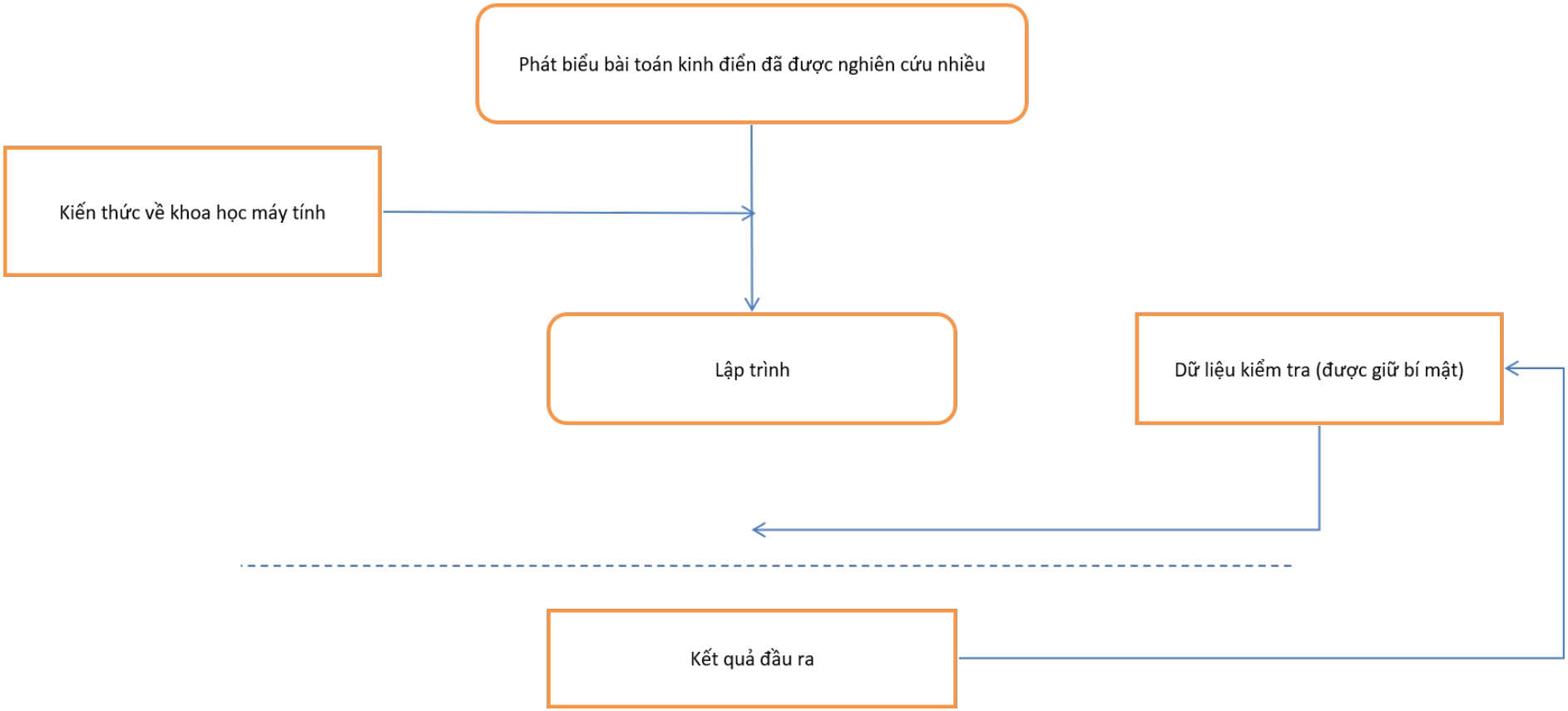
Đỗ Phan Thuận

thuandp.sinhvien@gmail.com

Bộ môn Khoa Học Máy Tính, Viện CNTT & TT,
Trường Đại Học Bách Khoa Hà Nội.

Ngày 5 tháng 2 năm 2020

Mô hình Bài tập lập trình Thuật toán ứng dụng



Yếu tố chính : giải bài toán nhanh nhất có thể!

Mục tiêu

- Cho một bài toán lập trình, ta sẽ phải
  - giải một cách hiệu quả,
  - sử dụng các thuật toán và cấu trúc dữ liệu,
  - chuyển lời giải thành chương trình,
  - làm càng nhanh càng tốt (dưới áp lực: thời gian, kết quả..),
  - và phải làm đúng (không sinh lỗi)

- Mục tiêu của bài giảng này là thực hành giải quyết những vấn đề trên.

Làm thế nào?

- Học những dạng bài phổ biến khác nhau
- Chỉ ra những ứng dụng của các thuật toán và cấu trúc dữ liệu bạn biết từ
  - khóa học cơ bản về các thuật toán
  - khóa học cơ bản về cấu trúc dữ liệu
- Giới thiệu các dạng thuật toán và cấu trúc dữ liệu phổ biến khác
- Học một số lý thuyết hay dùng
- Thực hành giải bài toán
- Thực hành lập trình
- Thực hành nữa
- .. và thực hành mãi

Sách tham khảo

- Competitive Programming by Steven Halim `http://libgen.io/ads.php?md5=f6f195012783a8b3c8bb7628882a51b7`
- Slides bài giảng Phân tích và thiết kế thuật toán Nguyễn Đức Nghĩa
- Bài giảng Chuyên đề Lê Minh Hoàng

Các bài giảng

- Dự kiến các chủ đề

Thứ tự.	Thời gian	Chủ đề	Hoạt động
1		Giới thiệu	
2		Cấu trúc dữ liệu và thư viện	
3		Thực hành	
4		Kỹ thuật đệ qui và nhánh cận	
5		Chia để trị	
6		Qui hoạch động	
7		Thực hành	
8		Thực hành và Kiểm tra giữa kỳ	
9		Qui hoạch động	
10		Đồ thị	
11		Thực hành	
12		Đồ thị	
13		Xử lý xâu	
14		Thực hành	
15		Lớp bài toán NP-đầy đủ	

Mẫu đề bài

- Mẫu chuẩn trong hầu hết các kỳ thi bao gồm
  - Mô tả bài toán
  - Mô tả định dạng dữ liệu vào
  - Mô tả định dạng kết quả ra
  - Ví dụ Dữ liệu vào/Kết quả ra
  - Giới hạn thời gian theo giây
  - Giới hạn bộ nhớ theo bytes/megabytes
- Yêu cầu viết chương trình giải bài toán đúng càng nhiều bộ dữ liệu càng tốt. Mặc định là dữ liệu vào không cần kiểm tra tính đúng đắn
- Chương trình không được chạy quá giới hạn thời gian và giới hạn bộ nhớ

Bài toán ví dụ

Mô tả bài toán

Viết chương trình nhân hai số nguyên.

Mô tả dữ liệu vào

Dòng đầu tiên chứa một số nguyên T, với $1 \leq T \leq 100$, là số lượng bộ test. T dòng tiếp theo, mỗi dòng chứa một test. Mỗi test bao gồm 2 số nguyên A, B, với $-2^{20} \leq A, B \leq 2^{20}$, cách nhau ít nhất một dấu cách.

Mô tả kết quả ra

Kết quả ghi ra mỗi dòng tương ứng với một test chứa một số là giá trị $A \times B$.

Bài toán ví dụ

Ví dụ dữ liệu vào	Dữ liệu kết quả ra
4 3 4 13 0 1 8 100 100	12 0 8 10000

Lời giải ví dụ

```cpp
#include <iostream>
using namespace std;
int main() {
    int T;
    cin >> T;
    for (int t = 0; t < T; t++) {
        int A, B;
        cin >> A >> B;
        cout << A * B << endl;
    }
    return 0;
}
```

Lời giải ví dụ

```
#include <iostream>
using namespace std;
int main() {
    int T;
    cin >> T;
    for (int t = 0; t < T; t++) {
        int A, B;
        cin >> A >> B;
        cout << A * B << endl;
    }
    return 0;
}
```

- Lời giải này có đúng không?

Lời giải ví dụ

```
#include <iostream>
using namespace std;
int main() {
    int T;
    cin >> T;
    for (int t = 0; t < T; t++) {
        int A, B;
        cin >> A >> B;
        cout << A * B << endl;
    }
    return 0;
}
```

- Lời giải này có đúng không?
- Điều gì xảy ra nếu $A = B = 2^{20}$?

Lời giải ví dụ

```
#include <iostream>
using namespace std;
int main() {
    int T;
    cin >> T;
    for (int t = 0; t < T; t++) {
        int A, B;
        cin >> A >> B;
        cout << A * B << endl;
    }
    return 0;
}
```

- Lời giải này có đúng không?
- Điều gì xảy ra nếu $A = B = 2^{20}$? Kết quả ra 0...

Lời giải ví dụ

```
#include <iostream>
using namespace std;
int main() {
    int T;
    cin >> T;
    for (int t = 0; t < T; t++) {
        int A, B;
        cin >> A >> B;
        cout << A * B << endl;
    }
    return 0;
}
```

- Lời giải này có đúng không? KHÔNG!
- Điều gì xảy ra nếu $A = B = 2^{20}$? Kết quả ra 0...

Lời giải ví dụ

- Khi $A = B = 2^{20}$, kết quả phải là 2^{40}

Lời giải ví dụ

- Khi $A = B = 2^{20}$, kết quả phải là 2^{40}
- Quá lớn với biến nguyên 32-bit, nên bị tràn số

Lời giải ví dụ

- Khi $A = B = 2^{20}$, kết quả phải là 2^{40}
- Quá lớn với biến nguyên 32-bit, nên bị tràn số
- Sử dụng biến nguyên 64-bit sẽ cho lời giải đúng

Lời giải ví dụ

```cpp
#include <iostream>
using namespace std;
int main() {
    int T;
    cin >> T;
    for (int t = 0; t < T; t++) {
        long long A, B;
        cin >> A >> B;
        cout << A * B << endl;
    }
    return 0;
}
```

Lời giải ví dụ

```cpp
#include <iostream>
using namespace std;
int main() {
    int T;
    cin >> T;
    for (int t = 0; t < T; t++) {
        long long A, B;
        cin >> A >> B;
        cout << A * B << endl;
    }
    return 0;
}
```

- Lời giải này có đúng không?

Lời giải ví dụ

```
#include <iostream>
using namespace std;
int main() {
    int T;
    cin >> T;
    for (int t = 0; t < T; t++) {
        long long A, B;
        cin >> A >> B;
        cout << A * B << endl;
    }
    return 0;
}
```

- Lời giải này có đúng không? ĐÚNG!

Lời giải ví dụ

```
#include <iostream>
using namespace std;
int main() {
    int T;
    cin >> T;
    for (int t = 0; t < T; t++) {
        long long A, B;
        cin >> A >> B;
        cout << A * B << endl;
    }
    return 0;
}
```

- Lời giải này có đúng không? ĐÚNG!
- Làm thế nào nếu giá trị A, B lớn nữa?

Lời giải ví dụ

```
#include <iostream>
using namespace std;
int main() {
    int T;
    cin >> T;
    for (int t = 0; t < T; t++) {
        long long A, B;
        cin >> A >> B;
        cout << A * B << endl;
    }
    return 0;
}
```

- Lời giải này có đúng không? ĐÚNG!
- Làm thế nào nếu giá trị A, B lớn nữa? XỬ LÝ SỐ LỚN!

Hệ thống chấm điểm tự động

- codeforces.com
- vn.spoj.com
- https://ru.kattis.com/

- Khi nộp bài giải lên bạn sẽ nhận được ngay phản hồi kết quả
- Có thể nộp bài giải với:
  - C
  - C++
  - Java
  - Python 2
  - Python 3
  - C#
  - Pascal
  - ...

Các phản hồi

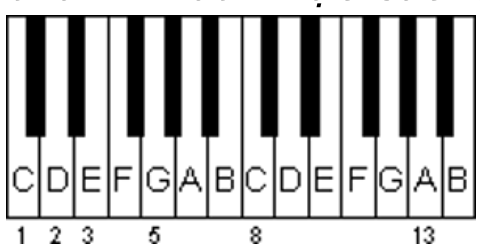
- Các phản hồi thường không thật chi tiết:
  - Accepted
  - Wrong Answer
  - Compile Error
  - Run Time Error
  - Time Limit Exceeded
  - Memory Limit Exceeded
- Có server cho phép chi tiết đến từng test, nhưng thông thường thì không

A. Kỹ năng đọc đề

- Kiến thức cơ sở của bài toán (Background): Đây là phần thể hiện Ứng dụng của thuật toán
- Các dữ kiện và yêu cầu của bài toán
- Mô tả khuôn dạng dữ liệu vào và ra
- Ví dụ khuôn dạng vào ra: thường là những trường hợp không quan trọng
- Các hạn chế cho các test của bài toán

Phần Background có quan trọng không?

- KHÔNG?
  - Ví dụ: *"Hãy lập trình sắp xếp n số thực. Hạn chế bộ nhớ: 3MB. Hạn chế thời gian: 1 giây. Hạn chế kích thước đầu vào $n \leq 10^6$. Hạn chế giá trị số thực trong 4 byte với định dạng đầu vào có chính xác hai chữ số sau dấu phẩy động"*.
  - Quá ngắn gọn và dễ hiểu!
- Tin học: BUỘC PHẢI CÓ!!!
  - Ví dụ: *Bản vanxơ Fibonacci là một bản nhạc mà giai điệu của nó bắt nguồn từ một trong những dãy số nổi tiếng nhất trong Lý thuyết số - dãy số Fibonacci. Hai số đầu tiên của dãy là số 1 và số 2, các số tiếp theo được xác định bằng tổng của hai số liên tiếp ngay trước nó trong dãy. Bản vanxơ Fibonacci thu được bằng việc chuyển dãy số Fibonacci thành dãy các nốt nhạc theo qui tắc chuyển một số nguyên dương thành nốt nhạc sau đây:*



Click here

  - Tạo hứng cho người đọc giải bài
  - Một chủ đề ứng dụng của KHMT

B. Kỹ năng phân loại bài toán

- Thực hành phân loại nhanh các dạng bài toán
- Có thể là kết hợp của nhiều dạng khác nhau

- Một số dạng bài hay gặp:

Loại	Loại hẹp
Ad Hoc	Trực tiếp
Ad Hoc	Mô phỏng
Duyệt toàn bộ	Lặp
Duyệt toàn bộ	Quay lui
Chia để & Trị	
Tham lam	Cổ điển
Tham lam	Mới
Quy hoạch động	Cổ điển
Quy hoạch động	Mới
Đồ thị	
Toán học	
Xử lý xâu	
Tính toán hình học	
Bài lạ	

C. Kỹ năng phân tích thuật toán

- Lời giải bài toán phải đủ nhanh và không sử dụng quá nhiều bộ nhớ
- Lời giải nên càng đơn giản càng tốt
- Sử dụng phương pháp Phân Tích Thuật Toán để xác định xem lời giải đưa ra có thỏa mãn giới hạn thời gian và bộ nhớ không
- Thông thường tính: 10^9 phép tính trong một giây
- Ví dụ cần sắp xếp $n \leq 10^6$ số nguyên chạy trong 3 giây
  - Liệu có thể sử dụng sắp xếp nổi bọt đơn giản $O(n^2)$ được không?
  - Sử dụng một thuật toán sắp xếp nhanh phức tạp hơn với $O(n \log n)$?
- Nếu cần sắp xếp $n \leq 10^3$ số nguyên chạy trong 3 giây
  - Bây giờ có thể sử dụng sắp xếp nổi bọt $O(n^2)$ được không?
- Luôn nhớ hãy sử dụng giải pháp đơn giản nhất thỏa mãn giới hạn thời gian

- Hãy thực hành cách ước lượng xấp xỉ trong đầu
- Mẹo: $2^{10} \approx 10^3$

- Trường hợp tìm ra lời giải mà bạn không chắc là nó đúng
- Hãy tìm cách chứng minh nó!
- Ngay cả khi không tìm được cách chứng minh hoặc phản bác nó thì điều thu được là hiểu sâu thêm bài toán

n	Độ phức tạp sát nhất	Ví dụ
≤ 10	$O(n!)$, $O(n^6)$	Liệt kê hoán vị
≤ 15	$O(2^n \times n^2)$	QHĐ cho bài TSP
≤ 20	$O(2^n)$, $O(n^5)$	QHĐ + Kỹ thuật bitmask
≤ 50	$O(n^4)$	QHĐ 3 chiều + $O(n)$ lặp, Tổ hợp ${}_nC_k = 4$
$\leq 10^2$	$O(n^3)$	Floyd Warshall
$\leq 10^3$	$O(n^2)$	Sắp xếp Nổi bọt/Chọn/Chèn
$\leq 10^5$	$O(n \log_2 n)$	Sắp xếp Trộn, Xây dựng Cây phân đoạn (Segment/Interval)
$\leq 10^6$	$O(n)$, $O(\log_2 n)$, $O(1)$	Thường đầu vào bài toán $n \leq 10^6$

D. Kỹ năng làm chủ ngôn ngữ lập trình

- Hãy thành thạo ngôn ngữ lập trình như lòng bàn tay
- Bao gồm cả các thư viện có sẵn
  - C++'s Standard Template Library
  - The Java Class Library
- Nếu đã có sẵn trong thư viện chuẩn thì không cần phải lập trình lại: cài đặt nhanh, không có bug
- Hạn chế : khả năng tùy biến của thư viện không linh hoạt. Rất nhiều phần kỹ thuật xử lý thuật toán không thể gọi trực tiếp thư viện mà phải tùy biến đi

E. Kỹ năng test chương trình

- Phải test để chắc chắn kết quả bài giải là đúng và thỏa mãn giới hạn thời gian
- Hoặc ít ra là biết lời giải sai nhưng không hiểu tại sao

- Cố gắng phản biện bài giải bằng cách tìm ra phản ví dụ (một dữ liệu vào mà bài giải trả kết quả ra sai, hoặc mất quá nhiều thời gian để tìm ra kết quả)
- Test các biên và dữ liệu lớn, ...
- Viết một thuật toán trực tiếp đơn giản để kiểm tra kết quả chương trình của mình có đúng không với những trường hợp kích thước đầu vào bé

F. Kỹ năng gõ nhanh

- Hãy trở thành thợ gõ nhanh và chính xác hơn
- Đừng để việc gõ lời giải là hạn chế cho việc giải bài
- Khi đánh máy không nhìn vào bàn phím, mắt nhìn vào màn hình kiểm tra luôn tính đúng đắn của việc gõ, trong lúc đó đầu vẫn có thể suy nghĩ song song các vấn đề tiếp theo

- TypeRacer là một cách luyện tập hiệu quả và thú vị:
- http://play.typeracer.com/

G: Thực hành, thực hành nữa và thực hành mãi

- Càng thực hành nhiều thì càng hoàn thiện các kỹ năng giải bài và lập trình
- Rất nhiều các trang chấm bài trực tuyến giúp bạn giải các bài toán của những kỳ thi trong quá khứ
- Một số trang giải bài trực tuyến thường xuyên tổ chức các cuộc thi đấu
- UVa, Codeforces, TopCoder, Kattis, ...

Lập trình viên thi đấu cũng như những vận động viên thể thao, cần phải rèn luyện thường xuyên để giữ được cảm hứng và phát triển các kỹ năng lập trình giải bài!

Các bài toán Ad Hoc

Các bài toán Ad Hoc

- Là dạng bài toán đơn giản nhất
- Thường làm đúng như mô tả bài toán yêu cầu
- Trực tiếp hoặc Mô phỏng
- Giới hạn thời gian thường không quan trọng
- Đôi khi mô tả dài dòng khó hiểu
- Đôi khi một số test biên lừa
- Một số bài toán phức hợp có thể khó lập trình

Bài toán: Cắt giảm chi tiêu

Công ty XYZ đang trong thời kỳ khủng hoảng và có nhiều khoản cần cắt giảm chi tiêu. Một số khoản là bỏ bớt không gian văn phòng, sử dụng mã nguồn mở, giảm thưởng, cắt giảm lương . . .

Họ có 3 nhân viên làm việc tại bộ phận kế toán và đang định cho nghỉ việc 2 người trong đó. Sau một loạt các cuộc họp, họ quyết định sa thải người đang nhận lương cao nhất và người đang nhận lương thấp nhất. Đây là quyết định thông thường khi có khủng hoảng xảy ra. Cho trước tiền lương của 3 nhân viên làm việc tại bộ phận kết toán. Yêu cầu đưa ra tiền lương của người được giữ lại.

Bài toán: Cắt giảm chi tiêu

Dữ liệu vào

Dòng đầu tiên là một số nguyên T ($T < 20$) là số lượng bộ test. Mỗi test bao gồm một dòng chứa 3 số nguyên dương phân biệt. 3 số này biểu diễn tiền lương của 3 nhân viên. Tất cả các số nguyên ở đây nằm trong khoảng $[1000, 10000]$.

Kết quả ra

Mỗi test đưa ra một số duy nhất là tiền lương của người được giữ lại.

Bài toán: Cắt giảm chi tiêu

Ví dụ dữ liệu vào	Ví dụ kết quả ra
3 1000 2000 3000 3000 2500 1500 1500 1200 1800	Case 1: 2000 Case 2: 2500 Case 3: 1500

Lời giải bài toán: Cắt giảm chi tiêu

```cpp
#include <cstdio>
#include <algorithm>
using namespace std;
int main() {
    int T;
    scanf("%d", &T);
    for (int t = 0; t < T; t++) {
        int salary[3];
        scanf("%d", &salary[0]);
        scanf("%d", &salary[1]);
        scanf("%d", &salary[2]);
        sort(salary, salary + 3);
        printf("Case %d: %d\n", t + 1, salary[1]);
    }
    return 0;
}
```

Bài toán: Gõ SMS

Điện thoại di động (ĐTDĐ) trở nên một phần không thể thiếu trong cuộc sống hiện đại. Ngoài việc gọi, ĐTDĐ có thể gửi tin nhắn mà người ta quen gọi là SMS. Không như bàn phím máy tính, đa phần ĐTDĐ hạn chế số phím. Để có thể gõ được tất cả các ký tự trong bảng chữ cái, nhiều ký tự sẽ được hiển thị trên cùng một phím. Vì vậy, để gõ một số ký tự, một phím sẽ phải được ấn liên tục đến khi ký tự cần tìm hiển thị trên màn hình.

Cho một đoạn văn bản, hãy tính số lần gõ phím để hiển thị được đoạn văn bản.

Bài toán: Gõ SMS

Bài toán giả thiết rằng các phím được sắp xếp như sau.

	abc	def
ghi	jkl	mno
pqrs	tuv	wxyz
	<SP>	

Trong bảng trên mỗi ô biểu diễn một phím. <SP> biểu diễn phím space. Để hiển thị ký tự 'a' thì sẽ phải bấm phím tương ứng 1 lần, nhưng để hiển thị ký tự 'b' của cùng phím đó thì sẽ phải bấm liên tục 2 lần và đối với phím 'c' là 3 lần. Tương tự, bấm 1 lần cho 'd', hai lần cho 'e' và 3 lần cho 'f'. Các ký tự khác cũng được làm tương tự. Lưu ý là để ra 1 khoảng trống thì cần bấm 1 lần phím space.

Bài toán: Gõ SMS

Dữ liệu vào

Dòng đầu tiên là một số nguyên T là số lượng bộ test. T dòng tiếp theo mỗi dòng chỉ chứa các khoảng trống và các ký tự in thường. Mỗi dòng chứa ít nhất 1 và tối đa 100 ký tự.

Kết quả ra

Mỗi test đầu vào tương ứng với một dòng ở kết quả ra. Mỗi dòng bắt đầu bởi thứ tự test và sau đó là một số biểu thị số lần bấm phím cho ra văn bản tương ứng. Xem ví dụ kết quả ra để thấy định dạng chuẩn xác.

Bài toán: Gõ SMS

Ví dụ dữ liệu vào	Ví dụ kết quả ra
2 welcome to ulab good luck and have fun	Case #1: 29 Case #2: 41

Lời giải bài toán: Gõ SMS

```
#include <cstdio>
#include <iostream>
#include <string>
using namespace std;
string keys[12] = {
    "",     "abc", "def",
    "ghi",  "jkl", "mno",
    "pqrs", "tuv", "wxyz",
    "",     " ",   ""
};
int main() {
    int T;
    scanf("%d\n", &T);
    for (int t = 0; t < T; t++) {
        // Each test case is handled here
    }
    return 0;
}
```

Lời giải bài toán: Gõ SMS

```
// Each test case:
string line;
getline(cin, line);
int cnt = 0;
for (int i = 0; i < line.size(); i++) {
    int cur;
    for (int j = 0; j < 12; j++) {
        for (int k = 0; k < keys[j].size(); k++) {
            if (line[i] == keys[j][k]) {
                cur = k + 1;
            }
        }
    }
    cnt += cur;
}
printf("Case #%d: %d\n", t + 1, cnt);
```