

Kỹ Thuật Random để giải quyết các bài toán có giới hạn nhỏ

— Tác giả - Nguyễn Ngọc Hiếu

— Wed, Jan 11, 2023 11:42 PM

Giới thiệu

Chắc hẳn bạn đã biết về giải thuật quay lui nhánh cận được sử dụng trong rất nhiều bài toán có giới hạn nhỏ ($n \leq 20$).

Tương tự với giải thuật quay lui trong các buổi train trước tôi đã giới thiệu cho các bạn về việc dùng hệ cơ số để giải quyết các bài toán chọn lựa. Kỹ thuật random mà hôm nay tôi muốn nhắc đến nó cũng không khác gì với việc sử dụng hệ cơ số. Tuy nhiên tôi nhận thấy rằng có một số bạn còn lười hay trong phòng thi cũng sẽ gặp vấn đề gì đó và có thể sẽ quên cách code 2 bài toán trên. Kỹ thuật Random sẽ giúp bạn giải quyết vấn đề đó.

- *Ưu điểm của giải thuật **ngắn gọn dễ nhớ** 100*, nhiều lúc ăn may sẽ chạy rất nhanh và chính xác, thậm chí bạn nào may mắn còn có thể giải quyết được mấy bài toán có giới hạn ($n \geq 20$)
- *Nhược điểm*
 - Không ổn định: Lúc thì random rất nhanh và đúng 👍, lúc thì chạy mãi không ra, lúc thì ra sai 👎
(Yên tâm đi thường thì đúng vẫn nhiều hơn mà với mấy bài nhỏ xíu thì "gần như" chắc chắn đúng).

Cài đặt

Ngôn ngữ C++

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  signed main(){
4      a = rand() // a đã được gán radom một số bất kỳ
5      // Nếu mà muốn random từ 0 đến n thì làm kiểu này
6      a = rand() % (n + 1)
7  }

```

Ngôn ngữ python

```

1  import random
2  a = random.randint(1, n) #a đã được gán random cho một số bất kỳ (1...n)

```

Ngôn ngữ C#

```

1  sing System;
2
3  public class Program{
4      public static void Main(){
5          Random rnd = new Random();
6          int a = rnd.Next(); // a đã được gán radom một số bất kỳ
7      // Nếu mà muốn random từ 0 đến n thì làm kiểu này
8          int a = rnd.Next(n + 1);
9      }
10 }

```

Ngôn ngữ js

```

1  //include random library
2  const random = require('random');
3
4  function main(){
5      let a = random.int(0, n);
6      // a đã được gán radom một số bất kỳ từ 0 đến n
7  }

```

Ngôn ngữ pascal

```
1  program Main;
2  var
3  a : integer;
4  begin
5  Randomize;
6  a := Random(n + 1);
7  end.
```

ý tưởng

Cho một bài toán chọn lựa, ứng với mỗi trường hợp ta có thể gán nó bằng một giá trị trong hệ cơ số rồi kiểm tra. ở đây trong gán giá trị ấy ta sẽ random để hạn chế thời gian chạy. Bởi, độ phức tạp thời gian của hàm random chỉ là $O(1)$. Với một dãy dài n phần tử thì độ phức tạp sẽ là $O(n)$. Tuy nhiên sẽ chẳng có ai ngu mà chỉ random một lần cả bởi xác suất đúng sẽ rất thấp vậy ta sẽ random với vòng lặp như khi random với hệ cơ số (Hoặc nhiều hơn) để tăng tỷ lệ đúng.

Ứng dụng

Ứng dụng của random cũng đã được cho vào đề thi hsg Hà Nội, tin học trẻ,...

I. Bài toán phát tiền thưởng(Trích HSG Hà Nội 2022)

Nhờ hoàn thành tốt công việc, An và Bình được công ty thưởng N tờ tiền. Tờ tiền thứ i có mệnh giá a_i . Hai bạn muốn chia đôi số tiền thành hai phần bằng nhau bằng cách chia cho mỗi người một số tờ tiền. Vì thế hai bạn quyết định sẽ chọn ra những tờ tiền để tổng số tiền hai bạn nhận được bằng nhau và lớn nhất, phần còn lại (nếu có) sẽ đem đi đầu tư.

Yêu cầu: Hãy giúp hai bạn tính tổng số tiền lớn nhất mà mỗi người nhận được trước khi đầu tư.

Dữ liệu vào

- Dòng đầu tiên chứa số nguyên dương $N(N \leq 500)$;
- Dòng thứ hai bao gồm N số nguyên dương a_1, a_2, \dots, a_N là mệnh giá của những tờ tiền.
- Tổng giá trị những tờ tiền sẽ không vượt quá 105.

Kết quả

- Gồm một dòng duy nhất là số tiền lớn nhất mà mỗi người nhận được.

Ràng buộc

- Có 40 số test ứng với 40 số điểm của bài thoả mãn $N \leq 3$;

- 30 số test tiếp theo ứng với 30 số điểm của bài thoả mãn $N \leq 12$;
- 30 số test còn lại ứng với 30 số điểm của bài không có ràng buộc gì thêm.

Ví dụ

Input

```
5
1 2 4 5 2
```

Output

```
7
```

Giải thích:

- An có thể chọn những tờ tiền có mệnh giá 1, 2, 4.
- Bình chọn những tờ tiền còn lại có mệnh giá 5, 2.
- Mỗi người sẽ nhận được tổng số tiền là 7. Vì số tiền mỗi người nhận đã bằng nhau và đã chia hết số tiền nên họ sẽ không đầu tư.

Input 2

Copy

```
5
9 8 4 5 13
```

Output

```
17
```

Giải thích:

- An sẽ chọn những tờ tiền có mệnh giá 9, 8.
- Bình sẽ chọn những tờ tiền có mệnh giá 4, 13.
- Mỗi người sẽ nhận được tổng số tiền là 17. Tờ tiền còn lại có mệnh giá 5 sẽ đem đi đầu tư.

Lời giải

Nhận thấy sẽ cho 2 trường hợp của một từ tiền bất kỳ là: An chọn, bình chọn và không ai chọn. Tin học hóa ta sẽ coi An chọn, Bình chọn và không ai chọn lần lượt là 0, 1, 2. Đọc đến đây ta đã nghĩ đến ngay việc sử dụng hệ tam phân giải quyết vấn đề. Đúng vậy ở mỗi mệnh giá tiền ta sẽ random 3 số 0 – 1 – 2.

Code mẫu cpp

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int n, a[867010];
5  signed main(){
6      freopen("ct.inp", "r", stdin);
7      freopen("ct.out", "w", stdout);
8      ios_base::sync_with_stdio(false);
9      cin.tie(0); cout.tie(0);
10     cin >> n;
11     for(int i = 0; i < n; i++)
12         cin >> a[i];
13     int res = 0;
14     for(int x = 0; x <= 60000; x++){
15         vector<int> v(n + 1, 0);
16         for(int i = 0; i < n; i++){
17             v[i] = (rand() + 1) % 3;
18         }
19         int t = 0, s = 0;
20         for(int i = 0; i < n; i++){
21             if(v[i] == 0){
22                 s += a[i];
23             }
24             if(v[i] == 1){
25                 t += a[i];
26             }
27         }
28         if(t == s)
29             res = max(res, t);
30     }
31
32     cout << res;
33 }
```

Hãy thử xem lại code dùng hệ cơ số

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  string covert(int K, int b)
4  {
5      string res;
6      while (K != 0)
7      {
8          res = (char)(K % b + 48) + res;
9          K /= b;
10     }
11     return res;
12 }
13 int n, a[5010];
14 signed main(){
15     freopen("ct.inp", "r", stdin);
16     freopen("ct.out", "w", stdout);
17     ios_base::sync_with_stdio(false);
18     cin.tie(0); cout.tie(0);
19     cin >> n;
20     for(int i = 0; i < n; i++)
21         cin >> a[i];
22     int res = 0;
23     for(int i = 1; i <= 1000000; i++){
24         string h = covert(i, 3);
25         int t = 0, s = 0;
26         if(h.size() <= n){
27             for(int i = 0; i < h.size(); i++){
28                 if(h[i] == '1')
29                     t += a[i];
30                 if(h[i] == '0')
31                     s += a[i];
32             }
33             if(t == s)
34                 res = max(res, t);
35         }
36     }
37
38     cout << res;
39 }
40

```

Dùng random để hiểu và để cài đặt hơn rất nhiều bạn nhĩ.

Chúc các bạn hiểu và sẽ mong nó sẽ giúp bạn trong học tập.

Bài tập luyện

Mấy bài này mình đều đã hướng dẫn trước đó bằng hệ cơ số rồi bạn có thể mở lại tài liệu buổi trước để xem lại ý tưởng và code nó bằng radom nhé.

I. Biểu Thức Cộng - Trừ

Cho một số nguyên không âm M và một biểu thức S chứa toàn các kí tự số từ 0 tới 9.

Yêu cầu: Hãy tìm một cách chèn các dấu $+$ và $-$ vào trước mỗi kí tự của S sao cho kết quả biểu thức thu được chính bằng số nguyên dương M cho trước.

Mô tả đầu vào

Dòng đầu là số nguyên không âm M .

Dòng tiếp theo là xâu kí tự S .

Ràng buộc

$$0 \leq M \leq 500 \leq M \leq 50.$$

$|S| \leq 10, |S| \leq 10$, với $|S|$ là độ dài của chuỗi S .

Mô tả đầu ra

In ra một cách chèn thỏa mãn. Nếu không có cách nào, in ra số -1 (không cần in ra các dấu cách phân tách toán hạng và toán tử).

Test case mẫu

Đầu vào

```
5
123456789
```

Đầu ra

```
-1-2-3+4-5+6+7+8-9
```

II. Biểu Thức Nhân - Cộng

Cho N số nguyên dương a_1, a_2, \dots, a_N cùng với 2 phép nhân và $N - 3$ phép cộng.

Yêu cầu: Hãy đặt 2 phép nhân và $N - 3$ phép cộng vào giữa các số a_i để thu được biểu thức có giá trị lớn nhất? Biết rằng không được phép đổi chỗ các số trong dãy.

Mô tả đầu vào

- Dòng đầu tiên chứa số nguyên dương N .
- N dòng tiếp theo, mỗi dòng chứa một số nguyên dương a_i là một số trong dãy.

Ràng buộc

$$4 \leq N \leq 1000.$$

$$1 \leq a_i \leq 10^4; \forall i : 1 \leq i \leq N$$

Mô tả đầu ra

Số nguyên duy nhất là giá trị lớn nhất của biểu thức tìm được.

Test case mẫu

Đầu vào mẫu 1

5
4
7
1
5
3

Đầu ra mẫu 1

44

Giải nghĩa: Biểu thức thu được là: $4 \times 7 + 1 + 5 \times 3 = 44$