PreVOI2022 Training

ĐH Vinh - Thanh Hoá - Hà Tĩnh - Ninh Thuận

dophanthuan@gmail.com

Khoa Học Máy Tính
Đại học Bách Khoa Hà Nội

Ngày 14 tháng 11 năm 2022

Bài 1. Đồng hồ báo thức

Bài 2. Thuê đường truyền

Bài 3. Mùa giáng sinh

Bài 1. Đồng hồ báo thức

Bài 2. Thuê đường truyền

Bài 3. Mùa giáng sinh

Lớp học — Class

Đồng hồ được cấu hình sẵn với K cuộc gọi đánh thức. Trước khi đi ngủ, người dùng lập trình đồng hồ với một mảng tham số A gồm các giá trị $A_1, A_2, ..., A_N$. Vào buổi sáng, đồng hồ sẽ đổ chuông K lần, với lần đổ chuông thứ i có độ mạnh P_i.
Để tính P_i, đồng hồ sẽ tạo tất cả các mảng con có các phần tử liên tiếp của mảng A và tính tổng lũy thừa bậc i của tất cả các mảng con này. Luỹ thừa bậc i của mảng con $A_j, A_{j+1}, ..., A_k$ được định nghĩa là $A_j \times 1^i + A_{j+1} \times 2^i + A_{j+2} \times 3^i + ... + A_k \times (k-j+1)^i$. Vì vậy P_i là tổng lũy thừa bậc i của tất cả các mảng con của mảng A.
Yêu cầu: Cho K và mảng A, hãy giúp Sơn tính tổng sức mạnh của mỗi lần báo thức: $P_1 + P_2 + ... + P_K$.

Subtask 1 ($N \leq 100, K \leq 20$): $O(N^3 \times K)$

Duyệt mọi mảng con của A và tính tổng $P_i, \forall i \leq K$.

```
result = 0
for(k in 1 to K) {
  for(L in 1 to N) {
    for(R in L to N) {
      for(j in L to R) {
        result = result + A[j] * pow(j-L+1,k)
        result %= 1000000007
      }
    }
  }
}
```

Tính trước tất cả các giá trị $pow(a, b)$ với $1 \leq a \leq n$ và $1 \leq b \leq k$.

Subtask 2 ($N \leq 10^6, K \leq 10^4$): $O(N \times \log(K))$

Duyệt mọi vị trí x và tính phần của A_x đóng góp vào kết quả cho tất cả các mảng con với phần tử này là phần tử thứ y trong mảng con.

- Nếu $y > x$, không có mảng con nào sao cho A_x là phần tử thứ y.
- Nếu $y \leq x$, có đúng một vị trí mà mảng con bắt đầu (nghĩa là $y - 1$ đứng trước x). Vì vậy, tất cả các mảng con bắt đầu từ $(n - (y - 1))$ và kết thúc vào hoặc sau vị trí x mà A_x ở vị trí y trong mảng con. Do đó, số mảng con có phần tử A_x ở vị trí thứ y trong mảng con sẽ là $(n - x + 1)$.

Subtask 2 ($N \leq 10^6, K \leq 10^4$): $O(N \times \log(K))$

Đóng góp từ A_x dưới dạng phần tử thứ y trong một mảng con $=$ $A_x \times y^1 + A_x \times y^2 + \ldots + A_x \times y^K$. Ký hiệu $S(x, y)$ là phần đóng góp từ A_x dưới dạng phần tử thứ y trong tất cả các mảng con. Ta có:

- $S(x, y) = (n - x + 1) \times A_x \times (y^1 + y^2 + \ldots + y^K)$.
- $S(x, y) = 0$ với $y > x$.
- $S(x, y) = A_x \times K \times (n - x + 1)$ với $y = 1$.
- $S(x, y) = \frac{(n-x+1)\times A_x \times y \times (y^K - 1)}{(y-1)}$ với $y \leq x$ và $y > 1$.

Subtask 2 ($N \leq 10^6, K \leq 10^4$): $O(N \times \log(K))$

Đóng góp của phần tử ở vị trí x vào kết quả là $C(x) = \sum S(x, y)$ với $1 \leq y \leq n$
$= (n - x + 1) \times A_x \times (K + \frac{2\times(2^K-1)}{(2-1)} + \frac{3\times(3^K-1)}{(3-1)} + \ldots + \frac{x\times(x^K-1)}{(x-1)})$.
Bước này có thể tính trong $O(N \times \log(K))$ với mỗi x, vậy tổng cộng là $O(N^2 \times \log(K))$ cho tất cả các phần tử. Vì vậy cần cải tiến như sau.

Subtask 2 ($N \le 10^6, K \le 10^4$): $O(N \times \log(K))$

Gọi $G(x) = \frac{C(x)}{(A_x \times (n-x+1))}$

$= K + \frac{2\times(2^K-1)}{(2-1)} + \frac{3\times(3^K-1)}{(3-1)} + \ldots + \frac{x\times(x^K-1)}{(x-1)}$.

Nhận xét:

$G(x+1) = G(x) + \frac{(x+1)\times((x+1)^K-1)}{x}$.

Vì vậy, có thể tính $G(x+1)$ từ $G(x)$ với thời gian $O(\log(K))$, rồi đến $C(x+1)$.

Do đó, tổng độ phức tạp $= O(N \times \log(K))$.

Subtask 2 ($N \leq 10^6, K \leq 10^4$): $O(N \times \log(K))$

```
G[1] = K
C[1] = A[1] * K * n
result = C[1]
for(i in 2 to n){
  G[i+1] = G[i] + i * (i^K - 1) / (i - 1)
  C[i] = G[i] * A[i] * (n - i + 1)

  result = result + C[i]
  result %= 1000000007
}
```

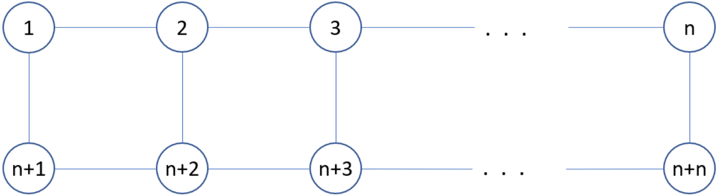
Bài 1. Đồng hồ báo thức

Bài 2. Thuê đường truyền

Bài 3. Mùa giáng sinh

Thuê đường truyền

Cho $2n$ máy tính $1, 2, ..., 2n$ được kết nối với nhau theo hình dưới:



Trong đó có các kết nối sau:

- Máy tính i được nối với máy $i+1$ với chi phí là u_i, $i = 1, 2, ..., n-1$;
- Máy tính i được nối với máy $i+1$ với chi phí là v_i, $i = n+1, n+2, ..., 2n-1$;
- Máy tính i được nối với máy tính $i+n$ với chi phí là w_i, $i = 1, 2, ..., n$.

Một phương án thuê đường truyền kết nối là một cách chọn ra các kết nối sao cho $2n$ máy tính có thể kết nối và truyền tin liên thông với nhau và tổng chi phí là số chẵn. Hãy tìm phương án thuê đường truyền với tổng chi phí là nhỏ nhất.

Subtask 1: $n \leq 5$

Duyệt toàn bộ.

Subtask 2: $n \leq 1000$; u_i, v_i, w_i là số chẵn

Tìm cây khung nhỏ nhất.

Subtask 3 ($n \leq 1000$): QHĐ $O(n)$

- Gọi $f[i][d]$ là tổng bé nhất với số dư d (0 hoặc 1) các cạnh bé nhất khi xét kết nối các đỉnh $1, 2, \ldots, i, n+1, n+2, \ldots, n+i$ tạo 1 cây khung.
- Tương tự với $g[i][d]$ nhưng chưa tạo được cây khung, 2 đỉnh i và $n+i$ chưa kết nối. Chỉ cần kết nối 2 đỉnh này nữa sẽ đảm bảo liên thông.
- Với trạng thái $f[i][d]$ ta có các cách chọn thêm các bộ cạnh $(a_i); (b_i); (a_i, b_i); (a_i, c_{i+1}); (b_i, c_{i+1})$. Mỗi cách tạo một trạng thái f hoặc g tương ứng.
- Với trạng thái $g[i][d]$ ta bắt buộc phải nối thêm 2 cạnh ra phía sau, nên ta có thêm bộ 2 cạnh (a_i, b_i) và có thể thêm (hoặc không thêm) cạnh c_{i+1} tạo trạng thái g hoặc f tương ứng.

Bài 1. Đồng hồ báo thức

Bài 2. Thuê đường truyền

Bài 3. Mùa giáng sinh

Mùa giáng sinh — Noel

Chính quyền thành phố Dresden đã mua N cây thông trang trí trên con đường lớn nhất của thành phố này. Họ sẽ tiến hành đặt N cây thông thành 2 hàng ở 2 bên của con đường. Một cách đặt sẽ khiến con đường lộng lẫy nếu như:

- Mỗi hàng đều phải có cây thông.
- Chênh lệch chiều cao giữa 2 cây thông liên tiếp trong cùng một hàng là như nhau và chiều cao của các cây sắp xếp theo thứ tự không giảm.

Yêu cầu: Hãy đề ra một phương án đặt các cây thông sao cho con đường trở nên lộng lẫy. Hoặc nếu không thể, hãy báo cáo lên chính quyền.

Subtask 1 ($N \leq 15$): $O(2^N)$

Thực hiện duyệt các dãy nhị phân, các vị trí có số 0 thì cây thông ở các vị trí đó sẽ cùng một hàng, tương tự với các vị trí có số 1. Sau khi có được các cây thông ở hai hàng, thực hiện sắp xếp không giảm và kiểm tra.

Subtask 2 ($N \leq 300$): $O(N^3)$

Thực hiện sắp xếp dãy $h_1, h_2, \ldots, h_N$ không giảm. Nhận thấy mỗi hàng cây đều phải là một dãy cấp số cộng. Ta có các trường hợp sau:

- Với $N \leq 4$ thì ta chia các cây vào 2 hàng sao cho mỗi hàng không quá 2 cây thông.
- Với $N > 4$ thì luôn tồn tại một hàng có nhiều hơn 2 cây thông, giả sử đó là hàng bên trái. Bởi vì hàng bên trái là một dãy cấp số cộng nên ta chỉ cần quan tâm 2 số đầu tiên của hàng đó. Ta duyệt 2 giá trị $h_i, h_j (i < j)$ là hai cây đầu tiên của hàng bên trái. Nếu $N - 2$ cây còn lại có thể tạo ra một dãy cấp số cộng thì thỏa mãn, nếu không thì ta thêm một cây có giá trị bằng $2 * h_j - h_i$ vào hàng bên trái và kiểm tra $N - 3$ cây còn lại có thỏa mãn hay không. Cứ như vậy, ta cứ tiến hành thêm 1 cây mới (nếu có tồn tại) vào hàng bên trái và kiểm tra các cây còn lại có thể tạo ra dãy cấp số cộng hay không.

Subtask 3 ($N \le 10^5$ và luôn tồn tại một phương án sao cho số lượng cây thông ở 2 hàng là như nhau)

Thực hiện sắp xếp dãy $h_1, h_2, \ldots, h_N$ không giảm. Ta xét cách giải quyết với $N > 4$ và chỉ có 3 trường hợp sau:

- Hai số đầu tiên của hàng bên trái là h_1, h_2.
- Hai số đầu tiên của hàng bên trái là h_1, h_3.
- Hai số đầu tiên của hàng bên trái là h_2, h_3.

Với mỗi trường hợp, ta sẽ tạo một dãy cấp số cộng gồm $N/2$ phần tử, và kiểm tra $N/2$ phần tử còn lại có tạo thành một dãy cấp số cộng hay không.

ĐPT: $O(N * 3)$.

Subtask 3 ($N \leq 10^5$): $O(N * 3 * \log N)$

Thực hiện sắp xếp dãy $h_1, h_2, \ldots, h_N$ không giảm. Ta thực hiện các trường hợp như ở Subtask 3, vấn đề của ta là phải kiểm tra rằng sau khi thêm 1 cây mới vào hàng bên trái thì các cây còn lại có thể tạo thành một dãy cấp số cộng hay không. Để giải quyết vấn đề này, ta sử dụng 2 multiset S và T. Multiset S để lưu chiều cao h_i của các cây còn lại, multiset T lưu chênh lệch chiều cao giữa 2 cây liên tiếp trong S.
Khi thêm một cây mới có chiều cao Y vào hàng bên trái thì:

- Xóa 1 giá trị Y trong multiset S.
- Đặt X là số liền trước Y, Z là số liền sau Y trong multiset S sau đó tiến hành xóa 2 giá trị $Y - X$ và $Z - Y$, đồng thời thêm giá trị $Z - X$ vào multiset T. Nếu tất cả các giá trị trong T bằng nhau (giá trị bé nhất bằng giá trị lớn nhất) chứng tỏ các cây còn lại có thể tạo thành 1 dãy cấp số cộng.