SUMMER CAMP 2022
Trường Chuyên Phan Bội Châu – Nghệ An

presented by Đỗ Phan Thuận
dophanthuan@gmail.com

Khoa Học Máy Tính
Đại học Bách Khoa Hà Nội

Ngày 19 tháng 5 năm 2022

CÁC CHỦ ĐỀ TỐI ƯU QUY HOẠCH ĐỘNG ĐÃ HỌC ĐỢT NÀY

- Quy hoạch động thuần tuý: cài đặt bằng vòng lặp hoặc đệ qui có nhớ
- Quy hoạch động truy vết
- Quy hoạch động trên mặt nạ bít
- Quy hoạch động + CTDL: dequeue, PQ, IT/BIT
- Quy hoạch động + Chia để trị
- Quy hoạch động + Đồ thị đặc biệt
- Quy hoạch động + Tham lam
- Biến đổi bài toán tối ưu về bài toán quyết định
- Chia để trị hình học

Bài 1. THREEJUG

Bài 2. REDSEQ

Bài 3. ARCHERY

Bài 4. FARM

Bài 5. VPER

Bài 6. PERDIFF

Bài 7. FACILITY

Bài 8. MEDPYRE

Bài 9. MEDPYRH

Bài 1. THREEJUG

Bài 2. REDSEQ

Bài 3. ARCHERY

Bài 4. FARM

Bài 5. VPER

Bài 6. PERDIFF

Bài 7. FACILITY

Bài 8. MEDPYRE

Bài 9. MEDPYRH

THREEJUG

Có 3 bình dung tích A, B, C (lít) với lượng nước ban đầu tương ứng là a, b, c (lít). Mỗi bước được phép đổ đúng d lít từ một bình i sang một bình j khác với điều kiện lượng nước hiện có trong bình i lớn hơn hoặc bằng d và sau khi đổ hết d lít từ bình i sang bình j thì nước trong bình j không bị tràn ra ngoài. Hãy tìm dãy ít nhất các bước đổ nước sao cho lượng nước còn lại ở 1 trong 3 bình đúng bằng T.

Thuật toán 60 điểm

Đúng với $A, B, C \leq 1000$.
Sử dụng phương pháp tìm kiếm theo chiều rộng BFS loang các trạng thái đổ có thể từ 3 bình. Do A, B, C đủ nhỏ nên có thể lưu hết được các trạng thái trong hàng đợi.

Thuật toán 100 điểm

$0 \leq A, B, C, a, b, c, d, T \leq 10^5$.
Bài toán đưa về giải 3 phương trình riêng biệt và đưa ra nghiệm nhỏ nhất trong 3 phương trình đó:
$a + dx = T$
$b + dy = T$
$c + dz = T$
Với mỗi phương trình, sử dụng phương pháp tham lam để thử đổ xem có thoả mãn điều kiện đề bài không.

Bài 1. THREEJUG

Bài 2. REDSEQ

Bài 3. ARCHERY

Bài 4. FARM

Bài 5. VPER

Bài 6. PERDIFF

Bài 7. FACILITY

Bài 8. MEDPYRE

Bài 9. MEDPYRH

Dãy số rút gọn — REDSEQ

Cho biết mẫu rút gọn của mỗi dãy số và một biểu thức, hãy cho biết dãy số là kết quả tính toán biểu thức.

60% số điểm

Dùng stack lưu trữ biểu thức và tính lần lượt.

100% số điểm

- Tiền xử lý bởi bảng logic bằng cách liệt kê tất cả các khả năng của biểu thức, có 2^n trường hợp. Mỗi dãy thành 1 phần tử logic, * thành AND và + thành OR. Sử dụng bitmask 16 bit để lưu trữ.
- Ví dụ: Kiểm tra x có thuộc $A * (B + C)$ đưa về kiểm tra biểu thức dạng $(b\ OR\ C) = 1$.
- Tiếp theo tính tất cả các phần tử của hợp tất cả biểu thức, sau đó kiểm tra xem một phần tử x có thuộc dãy số kết quả không bằng cách sử dụng kết quả tiền xử lý.
- Thay vì 16000*10 thì thay bằng 2^{16} tiền xử lý, phần còn lại giao nhau thì làm trực tiếp.
- Sử dụng CTDL bitset, hash về tối đa 160000 phần tử do $N \leq 16$, và số phần tử 1 dãy tối đa là 10000.

Bài 1. THREEJUG

Bài 2. REDSEQ

Bài 3. ARCHERY

Bài 4. FARM

Bài 5. VPER

Bài 6. PERDIFF

Bài 7. FACILITY

Bài 8. MEDPYRE

Bài 9. MEDPYRH

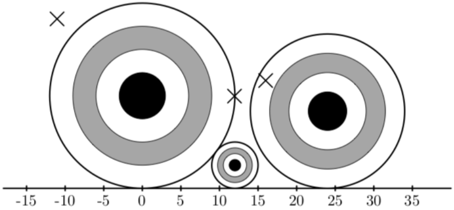
ARCHERY

Ta coi mục tiêu bắn có thể được biểu diễn dưới dạng mặt phẳng 2 chiều, trong đó $y = 0$ là mặt đất. Các mục tiêu có dạng vòng tròn, và tất cả các mục tiêu đều chạm mặt đất. Điều đó có nghĩa, nếu trung tâm của mục tiêu là (x, y) $(y > 0)$, thì bán kính của nó bằng y, để nó chạm vào dòng $y = 0$. Không có hai mục tiêu đồng thời có mặt tại tại bất kỳ thời điểm nào giao nhau (nhưng có thể tiếp xúc nhau).
Ban đầu không có mục tiêu bắn nào. Việc tham gia cuộc thi này có thể được mô tả là n sự kiện gồm 2 loại: hoặc sự kiện mục tiêu mới xuất hiện hoặc sự kiện vận động viên bắn mũi tên vào một điểm. Để đạt được mục tiêu, vận động viên phải bắn đúng bên trong vòng tròn (chạm vào đường biên không tính), khi đó mục tiêu đó sẽ bị xóa đi và vận động viên được thưởng một điểm.

Thuật toán 100 điểm

$1 \leq n \leq 2 \cdot 10^5$

- Với mỗi điểm, kẻ đường thẳng đi qua và vuông góc với trục x, đường thẳng này cắt tối đa $\log_2$(tọa độ tâm lớn nhất) (10^9) do tính chất hình học.
- Áp dụng cây IT quản lý các đoạn chiếu bởi hình tròn xuống trục.
- Độ phức tạp $O(n \log n \log c)$ ($\log c$ vì chia theo bán kính, $\log n$ vì xét tất cả $\log n$ đường tròn trong phạm vi). Phép kiểm tra một điểm có nằm trong đường tròn hay không chỉ mất $O(1)$.



Bài 1. THREEJUG

Bài 2. REDSEQ

Bài 3. ARCHERY

Bài 4. FARM

Bài 5. VPER

Bài 6. PERDIFF

Bài 7. FACILITY

Bài 8. MEDPYRE

Bài 9. MEDPYRH

FARM

Một trang trại trồng và cung cấp rau sạch ra thị trường cần lập kế hoạch sản xuất cho giai đoạn từ ngày 1 đến ngày n với tổng lượng hạt giống có để gieo trồng là Q. Do đặc tính thời vụ, nên khi gieo trồng 1 đơn vị hạt giống vào ngày i thì sẽ thu được một sản lượng là a_i. Kế hoạch sản xuất sẽ bao gồm các đợt gieo trồng, mỗi đợt sẽ cần tính toán gieo trồng một lượng hạt giống là bao nhiêu và vào ngày nào. Do đặc tính sinh trưởng và thu hoạch của rau nên 2 đợt trồng liên tiếp cách nhau ít nhất K ngày: cụ thể nếu đợt thứ nhất bắt đầu gieo trồng vào ngày thứ i thì đợt gieo trồng tiếp theo sẽ chỉ có thể thực hiện từ ngày $i + K$ trở đi. Ngoài ra, số đơn vị hạt giống gieo trồng trong mỗi đợt không vượt quá hằng số P cho trước.
Hãy tính toán kế hoạch sản xuất sao cho tổng sản lượng rau thu được là lớn nhất.

Thuật toán 50 điểm

$n, Q, P \leq 100$.
Độ phức tạp $O(n^4)$.

Thuật toán 70 điểm

$n, Q, P \leq 1000$.
Qui hoạch động 3 chiều theo n, Q, P. Độ phức tạp $O(n^3)$.

Thuật toán 100 điểm

$1 \le n \le 10^4, 1 \le K \le 10, 1 \le Q, P \le 10^4$;
$a_1, \ldots, a_n (1 \le a_i \le 10^3)$

- Bài toán có thể biến đổi thành tìm tập con có tối đa Q/P phần tử với tổng lớn nhất. Do đó có thể giảm bớt một chiều qui hoạch động theo chỉ số P.
- Gọi $dp(i, j)$ là sản lượng lớn nhất làm đến ngày i và đến đợt j.
  $dp(i, j) = MAX$ (

  $dp(i - 1, j)$, nếu ngày i không trồng,
  $dp(i - k, j - 1) + a[i]$, nếu trồng ngày i thì phải cách ra K ngày, $1 \le j \le R = Q/P$ là số đợt tối đa.
)
- Sau đó truy vết với sản lượng max thu được, đầy sản lượng mỗi ngày được chọn vào mảng sắp xếp tăng dần, và chọn tham lam từ sản lượng ngày lớn nhất trở về sao cho mỗi đợt không quá P và tổng không quá Q.

Bài 1. THREEJUG

Bài 2. REDSEQ

Bài 3. ARCHERY

Bài 4. FARM

Bài 5. VPER

Bài 6. PERDIFF

Bài 7. FACILITY

Bài 8. MEDPYRE

Bài 9. MEDPYRH

VPER

Cho dãy số nguyên a gồm n số nguyên dương: $a_1, a_2, \ldots, a_n$. Một dãy số được gọi là 1 hoán vị đúng từ 1 đến M khi và chỉ khi dãy đó chứa tất cả các số nguyên từ 1 đến M mỗi số đúng một lần. Bob có thể thực hiện hai thao tác: Xoá một số bất kì trong dãy và thêm một số bất kì vào cuối dãy. Bob thắc mắc là cần ít nhất bao nhiêu thao tác để biến dãy a thành hoán vị đúng.

Subtask 2: $1000 < n \leq 10^5$, $1 \leq S, a_i \leq 10^5$

- Xoá hết các số trùng nhau trong a
- Xét các giá trị của M từ 1 đến $2n+1$ và tính số biến đổi để thu được hoán vị M
- Gọi $f(M)$ là số lượng số bé hơn hoặc bằng M có trong dãy a, vậy ta phải thêm vào $M - f(M)$ và xóa đi $n - f(M)$ số
- $f(M)$ có thể tính dựa vào $f(M-1)$ bằng cách sắp xếp trước dãy a (có thể counting sort vì chỉ cần quan tâm đến $M <= 2n+1$)
- Sở dĩ ta chỉ cần xét đến $M <= 2n+1$ vì để xây dựng được hoán vị $2n+2$ cần thêm ít nhất là $n+2$ số, mà ta có thể xóa cả n số và thêm số 1 vào để được hoán vị $M = 1$ chỉ mất $n+1$ phép
- $res = Min(f(M)), M = 1, .., 2n+2$

Bài 1. THREEJUG

Bài 2. REDSEQ

Bài 3. ARCHERY

Bài 4. FARM

Bài 5. VPER

Bài 6. PERDIFF

Bài 7. FACILITY

Bài 8. MEDPYRE

Bài 9. MEDPYRH

PERDIFF

Đếm số lượng hoán vị của n số nguyên $1, 2, \ldots, n$ thoả mãn không tồn tại bất kỳ hai số liên tiếp nào trong mỗi hoán vị có trị tuyệt đối hiệu hai số bằng 1.
Gọi tính chất thoả mãn điều kiện đề bài là *tính tiêu chuẩn*

Subtask 1: $n \leq 10$

- Duyệt từng hoán vị và kiểm tra tính tiêu chuẩn
- ĐPT $O(n! \cdot n)$

Subtask 2: $n \leq 18$

- Gọi $dp[mask, i]$ là số lượng cách xây dựng hoán vị tiêu chuẩn từ tập con của tập $\{1, 2, \ldots, N\}$ tương ứng với giá tri nguyên $mask$ mà i là phần tử cuối cùng trong các hoán vị đang xét
- Với mỗi giá trị j thoả mãn $|j - i| > 1$ ta có

$$dp[mask|(1 << j), j] += dp[mask, i]$$

- ĐPT $O(2^n \cdot n^2)$

Subtask 3: $n \leq 1000$

Gọi $dp[i, j, t]$ là số lượng cách xây dựng i giá trị $1, 2, \ldots, i$ tạo nên j block tiêu chuẩn và t là số lượng vị trí có thể đặt một giá trị khác bên cạnh i

- $t = 0$: giá trị i kết nối hai block
- $t = 1$: giá trị i nằm ở biên, được thêm vào biên từ block có trước đó
- $t = 2$: giá trị i tạo một block mới

Ta có công thức QHĐ:

- $dp[i+1, j+1, 2] += dp[i, j, t]$: giá trị $i+1$ tạo block mới
- $dp[i+1, j, 1] += dp[i, j, t] \times (2j - t)$: giá trị $i+1$ xếp vào biên của block nào đó, có j block như vậy có $2j$ biên, loại trừ đi t vị trí không được xếp cạnh giá trị i
- $dp[i+1, j-1, 0] += dp[i, j, t] \times (j - t) \times (j - 1)$: giá trị $i+1$ kết nối hai block đã tạo trước đó, có $j(j-1)$ cặp block để có thể kết nối, mà có t vị trí i ở biên, nghĩa là có $t(j-1)$ cặp không thể kết nối bởi $i+1$, vậy còn lại $(j-t)(j-1)$ cặp kết nối được với nhau bởi $i+1$

ĐPT: $O(n^2)$

Subtask 4: $n \leq 10^6$

Gọi

- $f[n]$ là số lượng hoán vị tiêu chuẩn độ dài n
- $g[n]$ là số lượng hoán vị độ dài n mà có đúng một cặp trong $n-1$ cặp các giá trị liền kề nhau.
- $h[n]$ là số lượng hoán vị độ dài n mà giá trị $n-1$ đứng cạnh n

Ta sẽ chứng minh các công thức sau:

- $f[n] = (n-2)f[n-1] + g[n-1] - h[n-1]$
- $g[n] = 2(n-1)f[n-1] + 2g[n-1] + g[n-2]$
- $h[n] = 2f[n-1] + h[n-1]$
- Công thức rút gọn: $f[n] = (n+1)f(n-1) - (n-2)f[n-2] - (n-5)f[n-3] + (n-3)f[n-4]$

ĐPT: $O(n)$

Chứng minh: $f[n] = (n-2)f[n-1] + g[n-1] - h[n-1]$

- Đặt giá trị n vào các vị trí trong hoán vị tiêu chuẩn độ dài $n-1$ mà không đặt bên cạnh giá trị $n-1$ nên có $n-2$ cách chọn (trong tổng số n vị trí), do đó số lượng hoán vị như vậy bằng $(n-2)f[n-1]$
- Chèn vào giữa hai giá trị liên tiếp liền kề $(i, i+1)$ nào đó của hoán vị tiêu chuẩn độ dài $n-1$ trong đó $i \neq n-1$, số lượng hoán vị như vậy bằng $g[n-1] - h[n-1]$

Chứng minh: $g[n] = 2(n-1)f[n-1] + 2g[n-1] + g[n-2]$

Gọi $P[n, k]$ là số lượng hoán vị độ dài n mà giá trị k và giá trị $k+1$ xếp liền kề nhau (còn các cặp giá trị liên tiếp khác không xếp kề nhau). Khi đó ta có thể xem $(k, k+1)$ là một phần tử mang giá trị k trong hoán vị độ dài $n-1$ thỏa mãn điều kiện bài toán. Như vậy ta có:

- Nếu không có giá trị $k-1$ và $k+2$ xếp bên cạnh cặp $(k, k+1)$ thì số lượng hoán vị như vậy bằng $2f[n-1]$
- Nếu một trong hai giá trị $k-1$ và $k+2$ xếp bên cạnh cặp $(k, k+1)$ thì số lượng hoán vị như vậy bằng $P[n-1, k-1] + P[n-1, k]$
- Nếu cả hai giá trị $k-1$ và $k+2$ xếp bên cạnh cặp $(k, k+1)$ thì số lượng hoán vị như vậy bằng $P[n-2, k-1]$
- Vậy $P[n, k] = 2f[n-1] + P[n-1, k-1] + P[n-1, k] + P[n-2, k-1]$
- Cộng dồn các vị trí của k ta được đáp án: $g[n] = 2(n-1)f[n-1] + 2g[n-1] + g[n-2]$

Chứng minh: $h[n] = 2f[n-1] + h[n-1]$

- Nếu $n-2$ không xếp cạnh $n-1$ và sau đó ta sẽ đặt n cạnh $n-1$ thì số lượng hoán vị như vậy bằng $2f[n-1]$
- Nếu $n-2$ xếp cạnh $n-1$, sau đó ta chèn n vào giữa cặp $(n-2, n-1)$ thì số lượng hoán vị như vậy bằng $h[n-1]$

Bài 1. THREEJUG

Bài 2. REDSEQ

Bài 3. ARCHERY

Bài 4. FARM

Bài 5. VPER

Bài 6. PERDIFF

Bài 7. FACILITY

Bài 8. MEDPYRE

Bài 9. MEDPYRH

FACILITY

Một công ty cung cấp dịch vụ cho thuê kho chứa hàng. Công ty nhận được n đơn đặt thuê kho hàng của khách hàng $1, \ldots, n$, mỗi đơn thuê của khách hàng i sẽ bao gồm:

- s_i: ngày bắt đầu thuê
- d_i: số ngày cần thuê
- r_i: số tiền khách hàng i thuê phải trả cho công ty

Tại mỗi thời điểm, kho hàng của công ty chỉ có thể phục vụ cho 1 đơn thuê duy nhất, đồng thời khi một khách hàng kết thúc sử dụng kho hàng thì công ty cần có K ngày để bảo trì kho trước khi cho một khách hàng khác thuê: cụ thể, khách thứ nhất kết thúc thuê vào ngày thứ x thì khách thứ hai chỉ có thể thuê sau ngày thứ $x + K$. Hãy giúp công ty lựa chọn các khách để cho thuê sao cho tổng số tiền thu được là lớn nhất.

Thuật toán 30 điểm

Đúng với 30% số điểm có $n, K \leq 10$. Duyệt toàn bộ.

Thuật toán 80 điểm

Đúng với $n, K, s_i, d_i, r_i \leq 5 \times 10^4$

- Sort lại theo thời điểm kết thúc s
- Qui hoạch động: gọi $dp(i)$ là tổng số tiền thu được lớn nhất cho thuê đến ngày i:

dp(i) = MAX(

$dp(i-1)$ (nếu không cho thuê ngày i),

$dp(j) + r(i)$ với $j < i$, nếu cho thuê ngày i và ngày gần nhất cho thuê cách i ít nhất K ngày.

)

Độ phức tạp $O(n^2)$.

Thuật toán 100 điểm

$1 \leq n \leq 10^6, 1 \leq K \leq 10^9$
Lưu ý: Với những dữ liệu vào có số lượng lớn ký tự ví dụ khoảng $c \times 10^5$ đến $c \times 10^6$ thì nên sử dụng phương pháp đọc ghi dữ liệu nhanh (FastI/O).
Do $dp(i)$ tăng dần nên:

- Sử dụng phương pháp tìm kiếm nhị phân vị trí sát nhất có thời điểm kết thúc cách thời điểm bắt đầu của i ít nhất K ngày;
- hoặc BIT để tìm k tốt nhất trong đoạn trước i. Độ phức tạp $O(n \log n)$;

Phương pháp đọc ghi dữ liệu nhanh FastI/O

```cpp


template <typename T> inline void read(T &x){
    x = 0; char c;
    while (!isdigit(c = getchar()));
    do
        x = x * 10 + c - '0';
    while (isdigit(c = getchar()));
}

template <typename T> inline void write(T x){
    if (x > 9) write(x / 10);
    putchar(x % 10 + 48);
}

int main() {
    read(n);
    write(n);
}
```

Bài 1. THREEJUG

Bài 2. REDSEQ

Bài 3. ARCHERY

Bài 4. FARM

Bài 5. VPER

Bài 6. PERDIFF

Bài 7. FACILITY

Bài 8. MEDPYRE

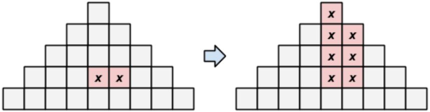
Bài 9. MEDPYRH

Kim tự tháp 1 — MEDPYRE

Khôi phục lại hoán vị tầng N của kim tự tháp khi biết giá trị x ở đỉnh. Mỗi giá trị trong 1 ô của kim tự tháp bằng trung vị của 3 số liền dưới.

100%

- $x = 1$ hoặc $2N - 1$, câu trả lời là "No". Tất cả các giá trị khác của x đều cho câu trả lời "Yes"
- Nhận xét: nếu trên hàng i có 2 ô liên tiếp j và $j + 1$ có cùng giá trị x thì tất cả các ô thứ j và $j + 1$ (nếu tồn tại) ở tất cả các hàng nhỏ hơn i cũng có giá trị x
- Vì vậy, chỉ cần đưa ra một hoán vị để tạo ra hai ô liên tiếp thứ N và $N + 1$ là x trên hàng thứ $N - 1$. Ví dụ:
  - nếu $x \neq 2$, một kết quả có thể là $(\ldots, x - 1, x, x + 1, x - 2, \ldots)$
  - nếu $x = 2$, một kết quả có thể là $(\ldots, x + 1, x, x - 1, x + 2, \ldots)$



Bài 1. THREEJUG

Bài 2. REDSEQ

Bài 3. ARCHERY

Bài 4. FARM

Bài 5. VPER

Bài 6. PERDIFF

Bài 7. FACILITY

Bài 8. MEDPYRE

Bài 9. MEDPYRH

Kim tự tháp 2 — MEDPYRH

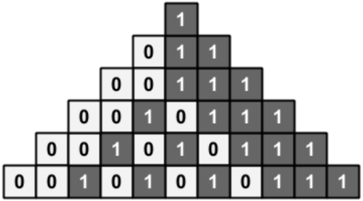
Khôi phục giá trị x ở đỉnh kim tự tháp khi biết hoán vị tầng N. Mỗi giá trị trong 1 ô của kim tự tháp bằng trung vị của 3 số liền dưới.

Subtask 1: $\sum N \leq 5000$

Thuật toán trực tiếp cho độ phức tạp N^2

Subtask 2: $A_i \in [0, 1]$ trong mọi bộ test

- Nếu có hai số 0 nằm kề nhau ở hàng dưới cùng, khi đó tất cả các giá trị trong hai cột này theo đường chéo kim tự tháp sẽ bằng 0. Điều tương tự với hai số 1 kề nhau ở hàng dưới cùng.
- Xét trường hợp khi số 0 và số 1 kề nhau. Xem hình sau:



- Khi có hai số 0 hoặc 1 giống nhau nằm liền kề theo chiều ngang, số này cũng xuất hiện dọc theo 2 cột đường chéo của kim tự tháp lên trên và và tạo thành hình chữ V
- Với những ô còn lại, giá trị của một ô là sự phủ định của giá trị ngay bên dưới nó
- ĐPT: $O(N)$

Subtask 3: $N \leq 10^5$

Tìm kiếm nhị phân:

- Tìm kiếm nhị phân giá trị x sử dụng $O(\log Q)$ câu hỏi, với Q là giá trị lớn nhất có thể, và trả lời câu hỏi: Giá trị ô trên đỉnh kim tự tháp ít nhất là x hay không?
- Với mỗi ô, cần quan tâm giá trị trong ô có ít nhất là x hay không
- thay thế tất cả các số nguyên nhỏ hơn x bằng 0 và các số nguyên khác bằng 1. Bài toán đưa về subtask 2
- ĐPT: $O(N \log Q)$