

Bài tập tương tự: Chặt nhị phân kết quả

Tài liệu này tổng hợp 8 bài tập cùng dạng với các ví dụ trong bài chặt nhị phân kết quả. Các đề được viết lại bằng tiếng Việt dựa trên các bài phổ biến ở LeetCode, SPOJ, CSES, GeeksforGeeks và dạng phỏng vấn thuật toán quốc tế.

Dấu hiệu nhận biết: đáp án là một số nguyên; có thể viết hàm `check(x)`; miền đúng/sai có tính đơn điệu. Với dạng tìm nhỏ nhất, khi `check(mid)=true` thì lưu đáp án và tìm bên trái. Với dạng tìm lớn nhất, khi `check(mid)=true` thì lưu đáp án và tìm bên phải.

Bộ test gồm 8 thư mục, mỗi bài 20 test, mỗi test có file `.inp` và `.out` tương ứng.

Bảng tổng hợp

Mã	Tên bài	Dạng đáp án	Khi <code>check(mid)=true</code>
BOOKSPLIT	Chia sách tối ưu	Tìm nhỏ nhất	<code>ans=mid, R=mid-1</code>
KOKO	Koko ăn chuối	Tìm nhỏ nhất	<code>ans=mid, R=mid-1</code>
AGGRCOW	Xếp bò xa nhau nhất	Tìm lớn nhất	<code>ans=mid, L=mid+1</code>
CUTROPE	Cắt dây thành K đoạn bằng nhau	Tìm lớn nhất	<code>ans=mid, L=mid+1</code>
FACTORY	Máy trong xưởng	Tìm nhỏ nhất	<code>ans=mid, R=mid-1</code>
WOODCUT	Cắt gỗ lấy đủ M đơn vị	Tìm lớn nhất	<code>ans=mid, L=mid+1</code>
SHIP	Chuyến hàng trong D ngày	Tìm nhỏ nhất	<code>ans=mid, R=mid-1</code>
SPLITARR	Chia dãy giảm tổng đoạn lớn nhất	Tìm nhỏ nhất	<code>ans=mid, R=mid-1</code>

Bài 1. BOOKSPLIT - Chia sách tối ưu

Nguồn gợi ý: Tương tự Book Allocation/Painter Partition trên các trang luyện thuật toán nước ngoài.

Đề bài

Có n quyển sách xếp theo thứ tự, quyển i có $a[i]$ trang. Cần chia dãy sách liên tiếp cho k học sinh. Mỗi quyển giao đúng một học sinh. Hãy tìm số trang lớn nhất của một học sinh nhỏ nhất có thể.

Dữ liệu vào

Dòng 1: n k . Dòng 2: n số $a[i]$.

Dữ liệu ra

Một số nguyên là giá trị nhỏ nhất của tổng trang lớn nhất.

Ràng buộc gợi ý

$1 \leq k \leq n \leq 2 \cdot 10^5$, $1 \leq a[i] \leq 10^9$.

Ý tưởng thuật toán

Chặt nhị phân trên đáp án $X =$ số trang tối đa cho mỗi học sinh. Nếu X càng lớn thì càng dễ chia. Tìm X nhỏ nhất thỏa mãn.

Hàm kiểm tra

$check(X)$: duyệt sách từ trái sang phải, gom vào học sinh hiện tại nếu tổng không vượt X ; nếu vượt thì mở học sinh mới. Nếu số học sinh cần dùng $\leq k$ thì đạt.

Độ phức tạp

$O(n \log(\text{sum } a))$.

Khung code C++

```
#include <bits/stdc++.h>
using namespace std;
using ll = long long;
bool check(const vector<ll>& a, int k, ll X){
    int cnt = 1; ll s = 0;
    for(ll v: a){
        if(s + v <= X) s += v;
        else { cnt++; s = v; }
    }
    return cnt <= k;
}
int main(){
    int n,k; cin>>n>>k; vector<ll>a(n);
    ll L=0,R=0;
    for(ll &x:a){ cin>>x; L=max(L,x); R+=x; }
    ll ans=R;
    while(L<=R){
        ll mid=(L+R)/2;
        if(check(a,k,mid)) ans=mid, R=mid-1;
        else L=mid+1;
    }
    cout<<ans;
}
```

Tóm tắt 20 test đã tạo

Test	Dòng đầu input	Output
1	4 2	113
2	4 2	60
3	4 4	5
4	1 1	7
5	9 3	17
6	5 2	102
7	6 3	23

8	3 1	180
9	10 9	78
10	10 7	77
11	11 2	232
12	15 5	135
13	5 3	58
14	14 11	78
15	12 8	76
16	8 2	177
17	8 2	185
18	11 11	63
19	7 6	106
20	6 6	61

Bài 2. KOKO - Koko ăn chuối

Nguồn gợi ý: Dựa trên dạng LeetCode 875 - Koko Eating Bananas.

Đề bài

Có n đống chuối, đống i có $a[i]$ quả. Mỗi giờ Koko chọn một đống và ăn tối đa X quả trong đống đó. Tìm tốc độ nguyên nhỏ nhất X để ăn hết trong không quá H giờ.

Dữ liệu vào

Dòng 1: n H . Dòng 2: n số $a[i]$.

Dữ liệu ra

Một số nguyên là tốc độ nhỏ nhất.

Ràng buộc gợi ý

$1 \leq n \leq 2 \cdot 10^5$, $n \leq H \leq 10^{18}$, $1 \leq a[i] \leq 10^9$.

Ý tưởng thuật toán

Chặt nhị phân trên X . X càng lớn thì tổng thời gian càng giảm. Tìm X nhỏ nhất đạt.

Hàm kiểm tra

$check(X)$: tổng giờ = $\sum(\text{ceil}(a[i]/X))$. Nếu tổng giờ $\leq H$ thì đạt. Trong C++: $\text{ceil}(a[i]/X) = (a[i]+X-1)/X$.

Độ phức tạp

$O(n \log(\max a))$.

Khung code C++

```
#include <bits/stdc++.h>
using namespace std;
using ll = long long;
bool check(const vector<ll>& a, ll H, ll X){
    ll hours = 0;
    for(ll v: a){
        hours += (v + X - 1) / X;
        if(hours > H) return false;
    }
    return true;
}
int main(){
    int n; ll H; cin>>n>>H; vector<ll>a(n);
    ll L=1,R=1; for(ll &x:a){ cin>>x; R=max(R,x); }
    ll ans=R;
    while(L<=R){
        ll mid=(L+R)/2;
        if(check(a,H,mid)) ans=mid, R=mid-1;
        else L=mid+1;
    }
    cout<<ans;
}
```

Tóm tắt 20 test đã tạo

Test	Dòng đầu input	Output
1	4 8	4
2	5 5	30
3	5 6	23
4	1 1	1
5	1 2	500000000
6	4 4	1
7	3 5	9
8	5 10	3
9	8 35	20

10	3 13	14
11	11 36	42
12	8 25	45
13	4 9	52
14	3 3	105
15	4 8	91
16	5 15	58
17	11 36	36
18	6 9	124
19	6 6	199
20	9 39	29

Bài 3. AGGRCOW - Xếp bò xa nhau nhất

Nguồn gợi ý: Dựa trên SPOJ AGGRCOW - Aggressive Cows.

Đề bài

Có n chuồng nằm trên trục số tại vị trí $a[i]$. Cần đặt k con bò vào k chuồng sao cho khoảng cách nhỏ nhất giữa hai con bò bất kỳ là lớn nhất.

Dữ liệu vào

Dòng 1: n k . Dòng 2: n vị trí $a[i]$.

Dữ liệu ra

Một số nguyên là khoảng cách nhỏ nhất lớn nhất có thể.

Ràng buộc gợi ý

$2 \leq k \leq n \leq 2 \cdot 10^5$, $0 \leq a[i] \leq 10^9$.

Ý tưởng thuật toán

Sắp xếp vị trí. Chặt nhị phân trên khoảng cách D . D càng lớn càng khó đặt. Tìm D lớn nhất đạt.

Hàm kiểm tra

$check(D)$: đặt bò đầu tiên ở chuồng nhỏ nhất, sau đó tham lam đặt con tiếp theo ở chuồng đầu tiên cách vị trí đặt gần nhất ít nhất D . Nếu đặt được $\geq k$ con thì đạt.

Độ phức tạp

$O(n \log(\text{range}))$.

Khung code C++

```
#include <bits/stdc++.h>
using namespace std;
using ll = long long;
bool check(vector<ll>& a, int k, ll D){
    int cnt=1; ll last=a[0];
    for(int i=1;i<(int)a.size();i++){
        if(a[i]-last>=D){
            cnt++; last=a[i];
            if(cnt>=k) return true;
        }
    }
    return cnt>=k;
}
int main(){
    int n,k; cin>>n>>k; vector<ll>a(n);
    for(ll &x:a) cin>>x; sort(a.begin(),a.end());
    ll L=0,R=a.back()-a.front(),ans=0;
    while(L<=R){
        ll mid=(L+R)/2;
        if(check(a,k,mid)) ans=mid, L=mid+1;
        else R=mid-1;
    }
    cout<<ans;
}
```

Tóm tắt 20 test đã tạo

Test	Dòng đầu input	Output
1	5 3	3
2	2 2	1
3	5 3	4
4	4 4	100
5	4 2	0
6	2 2	999999999
7	6 4	6
8	6 3	8

9	11 8	31
10	11 11	5
11	5 5	9
12	6 2	224
13	14 4	134
14	6 5	7
15	14 4	92
16	14 13	2
17	13 5	76
18	14 3	159
19	13 2	382
20	10 10	11

Bài 4. CUTROPE - Cắt dây thành K đoạn bằng nhau

Nguồn gợi ý: Tương tự bài Maximum length of K equal length ropes trên GeeksforGeeks và nhiều OJ.

Đề bài

Có n sợi dây nguyên độ dài $a[i]$. Có thể cắt các sợi dây thành nhiều đoạn. Hãy tìm độ dài nguyên lớn nhất L sao cho cắt được ít nhất k đoạn, mỗi đoạn dài đúng L.

Dữ liệu vào

Dòng 1: n k. Dòng 2: n độ dài $a[i]$.

Dữ liệu ra

Một số nguyên là L lớn nhất; nếu không thể có đoạn dương thì in 0.

Ràng buộc gợi ý

$1 \leq n \leq 2 \cdot 10^5$, $1 \leq k \leq 10^{18}$, $1 \leq a[i] \leq 10^9$.

Ý tưởng thuật toán

Chặt nhị phân trên L. L càng lớn thì số đoạn cắt được càng ít. Tìm L lớn nhất đạt.

Hàm kiểm tra

check(L): tổng số đoạn = $\sum(a[i]/L)$. Nếu tổng $\geq k$ thì đạt.

Độ phức tạp

$O(n \log(\max a))$.

Khung code C++

```
#include <bits/stdc++.h>
using namespace std;
using ll = long long;
bool check(const vector<ll>& a, ll k, ll L){
    ll pieces=0;
    for(ll v:a){
        pieces += v/L;
        if(pieces>=k) return true;
    }
    return false;
}
int main(){
    int n; ll k; cin>>n>>k; vector<ll>a(n);
    ll R=0; for(ll &x:a){ cin>>x; R=max(R,x); }
    ll L=1, ans=0;
    while(L<=R){
        ll mid=(L+R)/2;
        if(check(a,k,mid)) ans=mid, L=mid+1;
        else R=mid-1;
    }
    cout<<ans;
}
```

Tóm tắt 20 test đã tạo

Test	Dòng đầu input	Output
1	3 5	3
2	3 6	3
3	3 5	0
4	1 10	10
5	1 101	0
6	3 4	4
7	3 7	5
8	2 3	500000000
9	4 508	1

10	6 648	1
11	6 117	6
12	3 159	1
13	3 156	1
14	8 1306	1
15	9 307	2
16	8 559	1
17	7 459	2
18	11 444	3
19	3 206	1
20	8 270	4

Bài 5. FACTORY - Máy trong xưởng

Nguồn gợi ý: Dựa trên CSES Factory Machines.

Đề bài

Một xưởng có n máy, máy i cần $a[i]$ giây để làm một sản phẩm. Các máy làm song song. Hỏi thời gian nhỏ nhất để làm được ít nhất t sản phẩm.

Dữ liệu vào

Dòng 1: n t . Dòng 2: n số $a[i]$.

Dữ liệu ra

Một số nguyên là thời gian nhỏ nhất.

Ràng buộc gợi ý

$1 \leq n \leq 2 \cdot 10^5$, $1 \leq t \leq 10^{18}$, $1 \leq a[i] \leq 10^9$.

Ý tưởng thuật toán

Chặt nhị phân trên thời gian T . T càng lớn thì số sản phẩm làm được càng nhiều. Tìm T nhỏ nhất đạt.

Hàm kiểm tra

$check(T)$: tổng sản phẩm = $\sum(T/a[i])$. Nếu tổng $\geq t$ thì đạt. Cần chặn tràn số khi cộng.

Độ phức tạp

$O(n \log(\min(a) \cdot t))$.

Khung code C++

```
#include <bits/stdc++.h>
using namespace std;
using ll = long long;
bool check(const vector<ll>& a, ll t, ll T){
    __int128 made=0;
    for(ll v:a){
        made += T/v;
        if(made>=t) return true;
    }
    return false;
}
int main(){
    int n; ll t; cin>>n>>t; vector<ll>a(n);
    ll mn=LLONG_MAX; for(ll &x:a){ cin>>x; mn=min(mn,x); }
    ll L=0,R=mn*t,ans=R;
    while(L<=R){
        ll mid=(L+R)/2;
        if(check(a,t,mid)) ans=mid, R=mid-1;
        else L=mid+1;
    }
    cout<<ans;
}
```

Tóm tắt 20 test đã tạo

Test	Dòng đầu input	Output
1	3 10	12
2	1 10	10
3	3 3	5
4	1 2	2000000000
5	3 100	98
6	4 25	40
7	3 50	165
8	4 1000	500
9	3 513	2120

10	5 219	192
11	2 816	12129
12	6 32	48
13	9 975	750
14	3 800	7378
15	10 864	867
16	9 340	765
17	2 639	10512
18	9 444	920
19	3 670	6498
20	3 925	7980

Bài 6. WOODCUT - Cắt gỗ lấy đủ M đơn vị

Nguồn gợi ý: Tương tự dạng Binary Search in Forest/Wood Collection và bài EKO quen thuộc.

Đề bài

Có n cây, cây i cao h[i]. Đặt máy cưa ở độ cao X, phần cây cao hơn X sẽ bị cắt lấy gỗ. Tìm X lớn nhất sao cho thu được ít nhất M đơn vị gỗ.

Dữ liệu vào

Dòng 1: n M. Dòng 2: n số h[i].

Dữ liệu ra

Một số nguyên là độ cao cưa lớn nhất.

Ràng buộc gợi ý

$1 \leq n \leq 2 \cdot 10^5$, $0 \leq M \leq \sum(h[i])$, $1 \leq h[i] \leq 10^9$.

Ý tưởng thuật toán

Chặt nhị phân trên X. X càng cao thì lượng gỗ thu được càng ít. Tìm X lớn nhất vẫn đủ gỗ.

Hàm kiểm tra

check(X): wood = sum(max(0, h[i]-X)). Nếu wood >= M thì đạt.

Độ phức tạp

$O(n \log(\max h))$.

Khung code C++

```
#include <bits/stdc++.h>
using namespace std;
using ll = long long;
bool check(const vector<ll>& h, ll M, ll X){
    __int128 wood=0;
    for(ll v:h){
        if(v>X) wood += v-X;
        if(wood>=M) return true;
    }
    return false;
}
int main(){
    int n; ll M; cin>>n>>M; vector<ll>h(n);
    ll R=0; for(ll &x:h){ cin>>x; R=max(R,x); }
    ll L=0,ans=0;
    while(L<=R){
        ll mid=(L+R)/2;
        if(check(h,M,mid)) ans=mid, L=mid+1;
        else R=mid-1;
    }
    cout<<ans;
}
```

Tóm tắt 20 test đã tạo

Test	Dòng đầu input	Output
1	4 7	15
2	6 3	7
3	1 5	0
4	1 0	5
5	3 15	5
6	3 250	125
7	5 5	2
8	2 999999999	1
9	8 287	83

10	12 345	154
11	11 832	82
12	9 16	230
13	8 345	121
14	12 877	44
15	3 154	132
16	3 71	219
17	12 1946	4
18	4 194	144
19	3 132	125
20	12 156	210

Bài 7. SHIP - Chuyển hàng trong D ngày

Nguồn gợi ý: Dựa trên dạng LeetCode 1011 - Capacity To Ship Packages Within D Days.

Đề bài

Có n kiện hàng theo thứ tự, kiện i nặng $w[i]$. Mỗi ngày tàu chở một số kiện liên tiếp tiếp theo, tổng khối lượng không vượt quá sức chứa C. Tìm C nhỏ nhất để chuyển hết trong D ngày.

Dữ liệu vào

Dòng 1: n D. Dòng 2: n số $w[i]$.

Dữ liệu ra

Một số nguyên là sức chứa nhỏ nhất.

Ràng buộc gợi ý

$1 \leq D \leq n \leq 2 \cdot 10^5$, $1 \leq w[i] \leq 10^9$.

Ý tưởng thuật toán

Chặt nhị phân trên sức chứa C. C càng lớn thì số ngày cần dùng càng ít. Tìm C nhỏ nhất đạt.

Hàm kiểm tra

check(C): duyệt từ trái sang phải, xếp tối đa trong ngày hiện tại; nếu vượt C thì sang ngày mới. Nếu số ngày $\leq D$ thì đạt.

Độ phức tạp

$O(n \log(\text{sum } w))$.

Khung code C++

```
#include <bits/stdc++.h>
using namespace std;
using ll = long long;
bool check(const vector<ll>& w, int D, ll C){
    int days=1; ll s=0;
    for(ll x:w){
        if(s+x<=C) s+=x;
        else { days++; s=x; }
    }
    return days<=D;
}
int main(){
    int n,D; cin>>n>>D; vector<ll>w(n);
    ll L=0,R=0; for(ll &x:w){ cin>>x; L=max(L,x); R+=x; }
    ll ans=R;
    while(L<=R){
        ll mid=(L+R)/2;
        if(check(w,D,mid)) ans=mid, R=mid-1;
        else L=mid+1;
    }
    cout<<ans;
}
```

Tóm tắt 20 test đã tạo

Test	Dòng đầu input	Output
1	10 5	15
2	6 3	6
3	5 4	3
4	1 1	10
5	3 2	60
6	4 4	5
7	4 2	101
8	5 3	15
9	6 1	414

10	13 5	130
11	6 6	96
12	9 2	230
13	7 6	100
14	5 5	81
15	4 1	318
16	13 2	308
17	14 9	97
18	14 9	92
19	12 4	171
20	7 1	360

Bài 8. SPLITARR - Chia dãy giảm tổng đoạn lớn nhất

Nguồn gợi ý: Dựa trên dạng LeetCode 410 - Split Array Largest Sum.

Đề bài

Cho dãy n số nguyên dương $a[i]$. Chia dãy thành đúng k đoạn liên tiếp không rỗng. Hãy tối thiểu hóa tổng lớn nhất trong các đoạn.

Dữ liệu vào

Dòng 1: n k . Dòng 2: n số $a[i]$.

Dữ liệu ra

Một số nguyên là giá trị nhỏ nhất của tổng đoạn lớn nhất.

Ràng buộc gợi ý

$1 \leq k \leq n \leq 2 \cdot 10^5$, $1 \leq a[i] \leq 10^9$.

Ý tưởng thuật toán

Giống chia sách: chặt nhị phân trên X là tổng tối đa cho mỗi đoạn. X càng lớn càng dễ chia. Tìm X nhỏ nhất đạt.

Hàm kiểm tra

$check(X)$: tham lam tạo số đoạn ít nhất sao cho tổng mỗi đoạn $\leq X$. Nếu số đoạn $\leq k$ thì có thể chia thành đúng k đoạn bằng cách tách thêm các đoạn dương.

Độ phức tạp

$O(n \log(\text{sum } a))$.

Khung code C++

```
#include <bits/stdc++.h>
using namespace std;
using ll = long long;
bool check(const vector<ll>& a, int k, ll X){
    int cnt = 1; ll s = 0;
    for(ll v: a){
        if(s + v <= X) s += v;
        else { cnt++; s = v; }
    }
    return cnt <= k;
}
int main(){
    int n,k; cin>>n>>k; vector<ll>a(n);
    ll L=0,R=0;
    for(ll &x:a){ cin>>x; L=max(L,x); R+=x; }
    ll ans=R;
    while(L<=R){
        ll mid=(L+R)/2;
        if(check(a,k,mid)) ans=mid, R=mid-1;
        else L=mid+1;
    }
    cout<<ans;
}
```

Tóm tắt 20 test đã tạo

Test	Dòng đầu input	Output
1	5 2	18
2	5 2	9
3	3 3	4
4	1 1	5
5	5 3	9
6	6 5	4
7	3 2	200
8	6 3	2

9	7 7	99
10	12 6	161
11	6 5	73
12	4 4	77
13	6 6	76
14	11 10	96
15	15 4	185
16	12 10	98
17	6 4	99
18	14 5	160
19	8 8	85
20	15 7	133

Nguồn tham khảo nước ngoài

- LeetCode 875 - Koko Eating Bananas.
- SPOJ AGGRCOW - Aggressive Cows.
- CSES 1620 - Factory Machines.
- GeeksforGeeks - Maximum length of K equal length ropes.
- LeetCode 1011 - Capacity To Ship Packages Within D Days.
- LeetCode 410 - Split Array Largest Sum.
- Wood Collection / Binary Search in Forest style problems.