

Max even subsequence

- Given a sequence of n integers $a = a_1, \dots, a_n$. A subsequence of a consists of contiguous elements of a (for example, a_i, a_{i+1}, \dots, a_j). The weight of a subsequence is defined to be the sum of its elements. A subsequence is called even-subsequence if its weight is even. Find the even-subsequence of a having largest weight.
- **Input**
- Line 1: contains a positive integer n ($1 \leq n \leq 106$)
- Line 2: contains a_1, \dots, a_n ($-106 \leq a_i \leq 106$)
- **Output**
- The weight of the largest even-subsequence found, or write NOT_FOUND if no solution found.
- **Example**
- **Input**
- 8
- 4 -5 2 4 -8 2 3 1
- **Output**
- 6

Max even subsequence

Đề bài:

Cho một dãy số a gồm n số nguyên a_1, a_2, \dots, a_n .

Một đoạn con được định nghĩa là một đoạn các phần tử liên tiếp của dãy a .

Yêu cầu:

Tìm đoạn con có tổng các phần tử là chẵn và lớn nhất.

Max even subsequence – Dynamic Programming Algorithm

- Xác định bài toán con
 - $S0[i]$ là tổng của dãy con chẵn cực đại của dãy $a[1], \dots, a[i]$ mà phần tử cuối cùng là $a[i]$
 - $S1[i]$ là tổng của dãy con lẻ cực đại của dãy $a[1], a[2], \dots, a[i]$ mà phần tử cuối cùng là $a[i]$
- Điều kiện đầu
 - Nếu $a[1]$ chẵn thì $S0[1] = a[1]$ và $S1[1]$ không tồn tại
 - Nếu $a[1]$ lẻ thì $S1[1] = a[1]$ và $S0[1]$ không tồn tại
- Công thức quy hoạch động
 - Với mỗi i từ 2 đến n : tính đồng thời $S0[i]$ và $S1[i]$ phụ thuộc $S0[i-1]$, $S1[i-1]$, $a[i]$ tùy thuộc vào $a[i]$ chẵn hay lẻ.

Implementation – initialization

```
#include <bits/stdc++.h>
using namespace std;
const int N = 1e6+1;
int n;
long long a[N];
long long S0[N];
long long S1[N];
bool B0[N];
bool B1[N];
void input(){
    cin >> n;
    for(int i = 1; i <= n; i++) cin >> a[i];
}
```

Implementation – initialization

```
void solve(){
    if(a[1]%2==0){ S0[1] = a[1]; B0[1] = true; B1[1] = false; }else{
        S1[1] = a[1]; B1[1] = true; B0[1] = false;
    }
    for(int i = 2; i <= n; i++){
        if(a[i]%2 == 0){
            if(B0[i-1]){ if(S0[i-1] >0) S0[i] = S0[i-1] + a[i]; else S0[i] = a[i]; B0[i] = true; }
            else{ S0[i] = a[i]; B0[i] = true; }
            if(B1[i-1]){ S1[i] = S1[i-1] + a[i]; B1[i] = true; }
            else{ B1[i] = false; }
        }else{
            if(B1[i-1]){ S0[i] = S1[i-1] + a[i]; B0[i] = true; }
            else{ B0[i] = false; }
            if(B0[i-1]){ if(S0[i-1] > 0) S1[i] = S0[i-1] + a[i]; else S1[i] = a[i]; B1[i] = true; }
            else{ S1[i] = a[i]; B1[i] = true; }
        }
    }
    long long ans = -1e18;
    for(int i = 1; i <= n; i++) if(B0[i] && ans < S0[i]) ans = S0[i];
    cout << ans << endl;
}
```

Implementation – initialization

```
int main(){
    ios_base::sync_with_stdio(0); cin.tie(0);
    input();
    solve();
    return 0;
}
```