INC REPORT UNSOLVED C LC07

AOL Algorithm Design Analysis

Kelompok: C Level

Hanif Abdurrahman Ayash - 2702314611

Pilar Nalendra Sarwanto - 2702362604

Zenia Nadia Rifaniputri - 2702343832

UNIVERSITAS BINA NUSANTARA
JAKARTA
2024

Problem C

Double Chunks

You have a chocolate bar consisting of N chunks (numbered from 1 to N). Chunk i contains A_i peanut bits. You can divide the chocolate bar into several pieces, with each piece consisting of one or more consecutive chunks. Each chunk can only be part of one piece. The total number of peanut bits in a piece is simply the sum of the peanut bits from each of its chunks.

A piece is considered a *double chunk* if and only if it consists of exactly two chunks. You are required to divide the chocolate bar into as many double chunks as possible, all having the same total number of peanut bits. Determine the maximum number of double chunks you can get while satisfying this requirement.

Input

The first line consists of an integer N ($2 \leq N \leq 100\,000$).

The second line consists of N integers A_i ($1 \leq A_i \leq 10^9$).

Output

Output a single integer representing the maximum number of double chunks you can get while satisfying the requirement.

Sample Input #1

```
10
2 4 1 4 5 2 3 1 1 4
```

Sample Output #1

```
3
```

Explanation for the sample input/output #1

You can make 3 double chunks, each containing 5 peanut bits. The first piece consists of chunks 2 and 3, which have 4 bits and 1 bit, respectively. The second piece consists of chunks 6 and 7, which have 2 bits and 3 bits, respectively. The third piece consists of chunks 9 and 10, which have 1 bit and 4 bits, respectively.

Sample Input #2

```
7
1 2 1 1 1 2 1
```

Sample Output #2

```
2
```

Sample Input #3

```
5
1 2 3 4 5
```

Sample Output #3

```
1
```

Objektif soal:

Mencari jumlah maksimal pasangan 2 chink berurutan atau double chunks yang bisa dihasilkan dengan syarat setiap pasangan harus memiliki jumlah peanut yang sama dan dilarang overlap.

```c
#include <stdio.h>
#include <stdlib.h>

// Fungsi untuk menghitung jmlh double chunks coklat maksimal
int function_solusi(int N, long long* simpan) {
    if (N < 2) return 0;

    // Array untuk menyimpan semua kemungkinan jumlah dari dua chunk berurutan
    long long* sums = (long long*)malloc((N-1) * sizeof(long long));
    int sums_count = 0;

    // Hitung semua totalnya
    for (int i = 0; i < N-1; i++) {
        sums[i] = simpan[i] + simpan[i+1];
        sums_count++;
    }
```

Kami memulai dengan mencantumkan *library* yang dibutuhkan yakni stdio.h dan stdlib.h. Pada function_solusi, fungsi menerima parameter int N yang menjadi jumlah chunk dan array simpan untuk menyimpan jumlah peanut per-chunk. Jika jumlah chunk (N<2) maka akan mengembalikan 0 karena tidak mungkin ‘double’ chunk.

Selanjutnya, kita membuat array yang menyimpan semua kemungkinan jumlah dari 2 chuck berurutan serta menghitung semua totalnya.

```
    int max_chunks = 0;

    // Untuk setiap kemungkinan jumlah, coba buat double chunks sebanyak mungkin
    for (int i = 0; i < sums_count; i++) {
        long long current_sum = sums[i];
        int chunks = 0;
        int last_used = -2;  // biar gak overlap

        // Cek semua posisi untuk jumlah yang sama
        for (int j = 0; j < sums_count; j++) {
            if (sums[j] == current_sum && j > last_used + 1) {
                chunks++;
                last_used = j + 1;
            }
        }

        if (chunks > max_chunks) {
            max_chunks = chunks;
        }
    }

    free(sums);
    return max_chunks;
```

Kita akan menginisialisasikan maksimal chunk sebagai 0 terlebih dahulu. Kemudian, kita membuat loop untuk setiap kemungkinan jumlah. Chunks akan menghitung berapa *double chunk* yang sekiranya bisa dihasilkan dengan current_sum. last_used akan mencatat posisi terakhir supaya tidak overlap. Untuk setiap jumlah akan dicek berapa kali jumlah ini muncul di array sums. Kalau ditemukan jumlah yang sama dan tidak overlap maka akan increment chunk. max_chunks akan diupdate jika ditemukan jumlah yang menghasilkan double chunk lebih banyak.

```
int main() {
    int N;
    scanf("%d", &N);

    // Alokasi memori untuk array simpan
    long long* simpan = (long long*)malloc(N * sizeof(long long));

    // Baca input array simpan
    for (int i = 0; i < N; i++) {
        scanf("%lld", &simpan[i]);
    }

    // Hitung dan cetak hasil
    int result = solve_double_chunks(N, simpan);
    printf("%d\n", result);

    return 0;
}
```

Pada bagian ini adalah bagian *scanning* pada inputan user lalu memanggil function yang sebelumnya telah dibuat untuk memberikan output yang diinginkan.

Time Complexity:

- Loop luar menghasilkan O(N)
- Loop dalam menghasilkan O(N)
- Total Time Complexity dari loop: O(N^2)

Alasan tidak menemukan solusi saat pelaksanaan INC:

- Kurangnya komprehensi pada solusi yang diinginkan soal
- Kurangnya berlatih dalam menghadapi soal yang lebih bervariasi