



**Факультет программной инженерии и  
компьютерной техники**

Алгоритмы и структуры данных

**Лабораторная работа №2**

**Введение в алгоритмы**

Преподаватель: Косяков Михаил Сергеевич

Выполнил: Ле Чонг Дат

Группа: P3231

2021 г.

# 1 Задание

## Вариант: 1296. Гиперпереход

Дана последовательность чисел  $p_1, p_2, \dots, p_n$ . Найдите последовательную подпоследовательность с наибольшим итогом. Если сумма меньше 0, выведите 0.

# 2 Решение

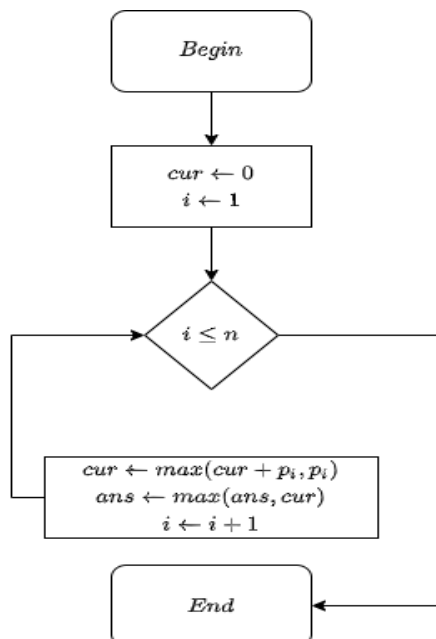
Вызов  $dp_i$ , чтобы быть максимальной суммой последовательных подпоследовательностей, заканчивающихся на  $i$ . Предполагая, что у нас есть значение  $dp_{i-1}$  (изначально  $dp_i = 0 \forall i$ ), учитывая  $i$ , есть два случая: последовательная подпоследовательность с наибольшей суммой продолжится до  $i$  или начнется из  $i \rightarrow dp_i = \max(dp_{i-1} + p_i, p_i)$

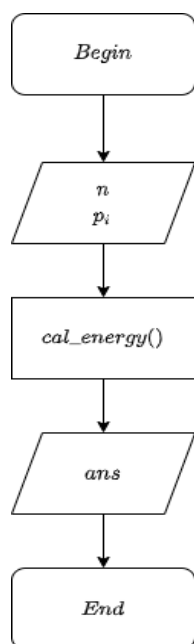
# 3 Схема программы

Глобальные переменные:

- $ans$  - Максимальное значение, которое нам нужно найти.
- $n$  - Количество камней.
- $w[i]$  - Масса  $i$ -го камня.

## 3.1 Подпрограмма cal\_energy()





## 4 Реализация

```
#include <bits/stdc++.h>
using namespace std;
const int N = 60010;
long long ans = 0;
int n;
int p[N];
void cal_energy() {
    long long cur = 0;
    for(int i = 1; i <= n; ++i) {
        cur = max(cur + p[i], (long long)p[i]);
        ans = max(ans, cur);
    }
}
int main() {
    ios::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    cin >> n;
    for(int i = 1; i <= n; ++i)
        cin >> p[i];
    cal_energy();
    cout << ans;
}
```

## 5 Вывод

Динамическое программирование - это в основном оптимизация простой рекурсии. Везде, где мы видим рекурсивное решение, которое повторяет вызовы одних и тех же входных данных, мы можем оптимизировать его с помощью динамического программирования. Идея состоит в том, чтобы просто сохранить результаты подзадач, чтобы нам не приходилось повторно вычислять их, когда это понадобится позже. Эта простая оптимизация сокращает временные сложности от экспоненциального до полиномиального.