Пограбування будинків

Ви - професіонал своєї справи і плануєте пограбувати ряд будинків уздовж вулиці. У кожному будинку захована певна сума грошей. Єдине, що заважає Вам грабувати - так це те, що сусідні будинки пов'язані системою безпеки: буде переданий сигнал в поліцію, якщо два сусідні будинки будуть пограбовані в один і той же вечір.

Знаючи кількість грошей в кожному будинку, визначте максимальну суму, яку Ви зможете пограбувати сьогодні ввечері без попередження поліції.

**Вхідні дані**

Перший рядок містить кількість будинків **n** (\*\*1 ≤ n ≤ **106**\*\*). Другий рядок містить **n** цілих невід'ємних чисел **a1,a2**, ...,**an**, де **ai**- кількість грошей, яке може бути винесено з **i** - го будинку.

**Вихідні дані**

Виведіть максимальну суму, яку Ви зможете пограбувати сьогодні ввечері без надходження сигналу в поліцію.



Ліміт часу **3** секунда

Ліміт використання пам'яті **128** MiB

**Вхідні дані #1**

5

6 1 2 10 4

**Вихідні дані #1**

16

**Ідея.**Грабіжнику потрібно вибрати такі будинки, щоб отримати максимум грошей. При цьому жодні два з таких будинків не повинні знаходитись поряд.  
Важливо зрозуміти, що порядок відвідування даних будинків не має значення, оскільки кінцевий результат — це сума всіх грошей, що знаходяться в них. Тому для отримання розв’язку, ми домовимось, що грабіжник буде ходити лише вперед, з початку вулиці в її кінець.

Пронумеруємо всі будинки від 0 до n-1.

“Герою” перш ніж відправитись на справу, потрібно визначити, скільки максимум грошей він зможе добути після відвідування кожного будинку. Для цього заведемо масив b, в який будемо записувати такі максимуми. Індекси даного масиву — номери відповідних будинків.

Для будинку 0, b[0]=a[0]. (Масив a — кількості грошей в будинках).  
Якщо відвідувати будинок 1, то будинок 0 прийдеться пропустити. Тому для будинку 1, b[1]=a[1].  
Якщо відвідувати будинок 2, то це доцільно зробити після відвідування будинку 0. Тому для будинку 2, b[2]=b[0]+a[2].

Визначимо, скільки можна отримати максимум грошей при відвідуванні будинку i.  
В будинок i можна потрапити, попередньо відвідавши або будинок i-2 або будинок i-3. З інших попередніх будинків в i-ий не вигідно іти, бо будуть пропуски. Тому для i-го будинку, b[i]=max(b[i-2], b[i-3])+a[i]

Очевидно, що останнім будинком подорожі може бути один із двох останніх будинків вулиці. Всі інші будинки не можуть бути останніми, бо будуть пропуски. Тому остаточний розв’язок залежить від максимуму грошей, які грабіжник може набрати після відвідування n-1-го або n-2-го будинків, тобто:  
res=max(b[n-1], b[n-2])

|  |  |
| --- | --- |
| Python | С++ |
| n=**int**(input())  a=**list**(map(int,input().**split**()))  b=[0]\*n  b[0] = a[0]  b[1] = a[1]  b[2] = b[0] + a[2]  **for** i **in** **range**(3,n):  b[i] = **max**(b[i - 2], b[i - 3]) + a[i]  **print**(max(b[n - 1], b[n - 2])) | #include <bits/stdc++.h>  using namespace std;  int main()  {  int n;  cin >> n;  int a[1000001] = {0};  long b[1000001] = {0};    for (int i = 0; i < n; i++) {  cin >> a[i];  }  b[0] = a[0];  b[1] = a[1];  b[2] = b[0] + a[2];  for (int i = 3; i < n; i++) {  b[i] = max(b[i - 2], b[i - 3]) + a[i];  }  long res = max(b[n - 1], b[n - 2]);    cout << res << endl;  return 0;  } |

<https://medium.com/@alexanderkuzmenko_86217/e-olymp-115-%D0%B4%D0%B2%D1%96-%D1%86%D0%B8%D1%84%D1%80%D0%B8-18d2de0bd5db>