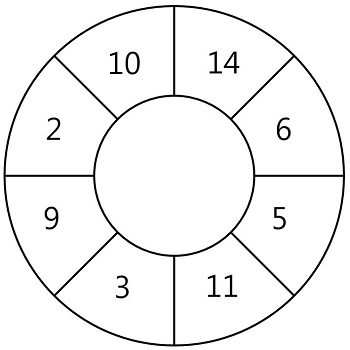
문제 설명

N개의 스티커가 원형으로 연결되어 있습니다. 다음 그림은 N = 8인 경우의 예시입니다.  
  
원형으로 연결된 스티커에서 몇 장의 스티커를 뜯어내어 뜯어낸 스티커에 적힌 숫자의 합이 최대가 되도록 하고 싶습니다. 단 스티커 한 장을 뜯어내면 양쪽으로 인접해있는 스티커는 찢어져서 사용할 수 없게 됩니다.

예를 들어 위 그림에서 14가 적힌 스티커를 뜯으면 인접해있는 10, 6이 적힌 스티커는 사용할 수 없습니다. 스티커에 적힌 숫자가 배열 형태로 주어질 때, 스티커를 뜯어내어 얻을 수 있는 숫자의 합의 최댓값을 return 하는 solution 함수를 완성해 주세요. 원형의 스티커 모양을 위해 배열의 첫 번째 원소와 마지막 원소가 서로 연결되어 있다고 간주합니다.

제한 사항

* sticker는 원형으로 연결된 스티커의 각 칸에 적힌 숫자가 순서대로 들어있는 배열로, 길이(N)는 1 이상 100,000 이하입니다.
* sticker의 각 원소는 스티커의 각 칸에 적힌 숫자이며, 각 칸에 적힌 숫자는 1 이상 100 이하의 자연수입니다.
* 원형의 스티커 모양을 위해 sticker 배열의 첫 번째 원소와 마지막 원소가 서로 연결되어있다고 간주합니다.

입출력 예

| **sticker** | **answer** |
| --- | --- |
| [14, 6, 5, 11, 3, 9, 2, 10] | 36 |
| [1, 3, 2, 5, 4] | 8 |

입출력 예 설명

입출력 예 #1  
6, 11, 9, 10이 적힌 스티커를 떼어 냈을 때 36으로 최대가 됩니다.

입출력 예 #2  
3, 5가 적힌 스티커를 떼어 냈을 때 8로 최대가 됩니다.

이 문제는 결국 풀지못해서 답을 찾아버렸다.

왜 이렇게 알고리즘 돌렸는지 이해는 했다.

http://blog.naver.com/meeting0103/221091798942

#include <vector>

#define max(a,b) ((a>b)?(a):(b))

using namespace std;

int solution(vector<int> sticker)

{

int answer = 0;

int size = 4; //배열을 이동시킬 횟수

int i, j;

if (sticker.size() == 1) return sticker[0]; // 배열이 한 개일 경우 바로 종료

while (size--)

{

int result[100000] = { 0, };

int tmp\_max = 0;

result[0] = sticker[0];

result[1] = sticker[1];

result[2] = sticker[2] + result[0];

// 현재 위치에서 2번째 전과 3번째 전 중에 큰 값을 더하여 현재값 계산

for (i = 3; i < sticker.size() - 1; i++)

{

tmp\_max = max(result[i - 2], result[i - 3]);

result[i] = sticker[i] + tmp\_max;

}

// 배열에서 마지막 계산값에서 최대값을 답으로 저장

answer = max(answer, result[sticker.size() - 2]);

// 배열을 이동시킴

sticker.push\_back(sticker[0]);

sticker.erase(sticker.begin());

}

return answer;

}

/\* 과정 설명

sticker[] = {14, 6, 5, 11, 3, 9, 2, 10}

result[] = {0, }

result[] = {14, 6, 14 + 5};

size - 1 = 7

i = 3

tmp\_max = max(result[1번] 6, 0번 14); tmp max = 14;

result[3] = 3번째 11 + 14;

result[] = {14, 6, 19, 25};

size = 7

i = 4

tmp\_max = max(result[2번] 19, 1번 6) tmp max = 19;

result[4] = 4번째 3 + 19

result[] = {14, 6, 19, 25, 22};

size = 7

i = 5

tmp\_max = max(result[3번] 25, 2번 19) tmp max = 25;

result[5] = 5번째 9 + 25

result[] = {14, 6, 19, 25, 22, 34};

size = 7

i = 6

tmp\_max = max(result[4번] 22, 3번 34) tmp max = 34;

result[6] = 6번째 2 + 34

result[] = {14, 6, 19, 25, 22, 34, 36};

answer =

\*/