The painter’s partition problem

> 给定一个boards数组，boards[i]表示画板i的长度也代表了填满画板所需要的时间；一共有K名画家， 每名画家可以在连续的画板上进行绘制，求在K名画家的情况下，绘制完所有画板所需的最短时间

```
#include <bits/stdc++.h>

using namespace std;

using vi = vector<int>;

// 在每位painter只能绘制maxLen长度的限制下，绘制boards数组所需要的painter数量
int numberOfPainters(vi &boards, int maxLen) {
    int total = 0, numberPainters = 1;
    for (int a: boards) {
        total += a;
        if (total > maxLen) {
            total = a;
            numberPainters++;
        }
    }
    return numberPainters;
}

int minPaintTime(vi &boards, int K) {
    // 此时对应painters = N的情况
    int minLen = *max_element(boards.begin(), boards.end());
    // 此时对应painters = 1的情况
    int maxLen = accumulate(boards.begin(), boards.end(), 0);
    // 在minTime与maxTime之间进行二分搜索painters = K的情况
    int low = minLen, high = maxLen;
    while (low < high) {
        int mid = low + (high - low) / 2;
        // 在每位painter最多绘制mid长度的限制下，绘制boards所需要的最小painter的数量
        int requiredPainters = numberOfPainters(boards, mid);
        if (requiredPainters <= K) high = mid;
        else
            low = mid + 1;
    }
    return low;
}

int main() {
    int T;
    scanf("%d", &T);
    while (T--) {
        int N, K;
        scanf("%d %d", &N, &K);
        vi boards(N);
        for (int i = 0; i < N; ++i) scanf("%d", &boards[i]);
        printf("%d\n", minPaintTime(boards, K));
    }
    return 0;
}
```