

Sep 04, 12 18:10

intervalSum.cpp

Page 1/1

```

/* FILE: intervalSum.cpp    last change: 23-July-2012    author: Romeo Rizzi
 * a solver for problem 1 (IntervalSum) in the 23-07-2012 exam in Algorithms
 */

// #include <iostream> // NOTA: ho commentato questa riga per essere sicuro di n
on aver lasciato operazioni di input/output di debug nella fretta di consegnare,
se compila cosi' non ci sono ed io non perdo stupidamente i miei punti-esame.
#include <cstdlib>
#include <fstream>
#include <cassert>

using namespace std;

const int MAX_N = 1000000;
int n, k, vect[MAX_N];

int maxSumFrom[MAX_N]; /* maxSumFrom[i] will store the maximum value of an int
erval starting in position i and of length at least k. */

int main() {
    ifstream fin("input.txt"); assert(fin);
    fin >> n >> k;
    for(int i = 0; i < n; i++)
        fin >> vect[i];
    fin.close();

    int best_so_far;
    int sum_of_k_from_pos, sum_all_from_pos = 0; // think of pos = n;
    if (k == 0) best_so_far = sum_of_k_from_pos = 0;
    for(int pos = n-1, length = 1; pos >= 0; pos--, length++ ) { // length = the
length of the interval from <pos> till the end of vect
        sum_all_from_pos += vect[pos];
        if(length == k)
            best_so_far = maxSumFrom[pos] = sum_of_k_from_pos = sum_all_from_pos;
        if(length > k) {
            sum_of_k_from_pos += vect[pos] - vect[pos+k];
            best_so_far = max( best_so_far,
                               maxSumFrom[pos] = max( sum_of_k_from_pos,
                                                         vect[pos] + maxSumFrom[pos+1] )
                               );
        }
    }

    ofstream fout("output.txt"); assert(fout);
    fout << best_so_far;
    fout.close();
    return 0;
}

```