# intervalloSommaMassimaDivideEtImpera.cpp

```
/* FILE: intervalloSommaMassimaDivideEtImpera.cpp   last change: 6-March-2012   author: Romeo Rizzi
  soluzione O(n log n) basata su tecnica divide et impera
 */

// #include <iostream> // NOTA: ho commentato questa riga per essere sicuro di non aver lasciato ope
razioni 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;

int n; const int MAX_N = 10000000; //numero elementi del vettore in input.txt
int vect[MAX_N]; //il vettore dell'istanza in input.txt

long int maxSumInterval(int first, int last) {
  /* ritorna il massimo valore di somma degli elementi
     di un sottointervallo dell'intervallo:
         vect[first ... last]
  */
  assert( last >= first );
  if( last == first ) return vect[first];
  assert( last > first );
  int mezz = (first + last) /2; assert( first <= mezz ); assert( mezz < last );
  long int opt = maxSumInterval(first, mezz);
  opt = max( opt, maxSumInterval(mezz+1, last) );

  int right_walker = mezz+1;
  long int right_wing, best_right_wing = right_wing = vect[right_walker];
  while( ++right_walker <= last ) {
    right_wing += vect[right_walker];
    best_right_wing = max(best_right_wing, right_wing);
  }
  // nota: l'ala destra vuota non e' stata considerata, quindi possiamo consentire all'ala sinistra
di essere vuota senza includere la soluzione intervallo vuoto
  long int left_wing, best_left_wing = left_wing = 0;
  for( int left_walker = mezz; left_walker >= first; left_walker-- ) {
    left_wing += vect[left_walker];
    best_left_wing = max(best_left_wing, left_wing);
  }
  opt = max( opt, best_left_wing + best_right_wing );
  return opt;
}


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

  ofstream fout("output.txt"); assert(fout);
  fout << maxSumInterval(0, n-1) << endl;
  fout.close();
  return 0;
}
```