ASD Lab 2

Alessandra Giordani

10-10-2014

Calendario

03/10	14:00-16:00	Introduzione
10/10	14:00-15:45	Ad-Hoc
17/10	(Nessuna lezione)	
24/10	14:00-16:00	Grafi 1
31/10	14:00-16:00	Grafi 2
07/11	14:00-16:00	Progetto 1
14/11	14:00-16:00	Progetto 1
21/11	14:00-16:00	Dinamica 1
28/11	14:00-16:00	Dinamica 2
05/12	14:00-16:00	Progetto 2
12/12	14:00-16:00	Progetto 2
19/12	(Nessuna lezione)	

Progetti: 06-15/11, 11-20/12
Iscrivetevi su
`http://goo.gl/NPqf4j`

Soluzioni: Sottosequenza di somma massima

Soluzioni presenti sulle prime slides del prof. Montresor

Idea di soluzione alternativa $O(N^2)$

Costruiamo array delle somme:

$$S_i = \sum_{j=1}^{i} A_j$$

Per ogni sottosequenza, calcoliamo la somma in $O(1)$:

$$\text{Somma da i a j} = S_j - S_{i-1}$$

	1	2	3	4	5
Array	2	-3	4	1	5
Somma	2	-1	3	4	9

Soluzioni: Sottomatrice di somma massima

Soluzione ovvia $(RC)^3$

- Ci sono $(RC)^2$ sottomatrici
- É possibile calcolare la somma di una sottomatrice in meno di $O(RC)$?
- Dobbiamo veramente guardare tutte le sottomatrici?

Soluzioni: Sottomatrice di somma massima

Calcolare la somma in $O(1)$

Stessa idea di prima. Riempiamo un array somma ($O(RC)$)

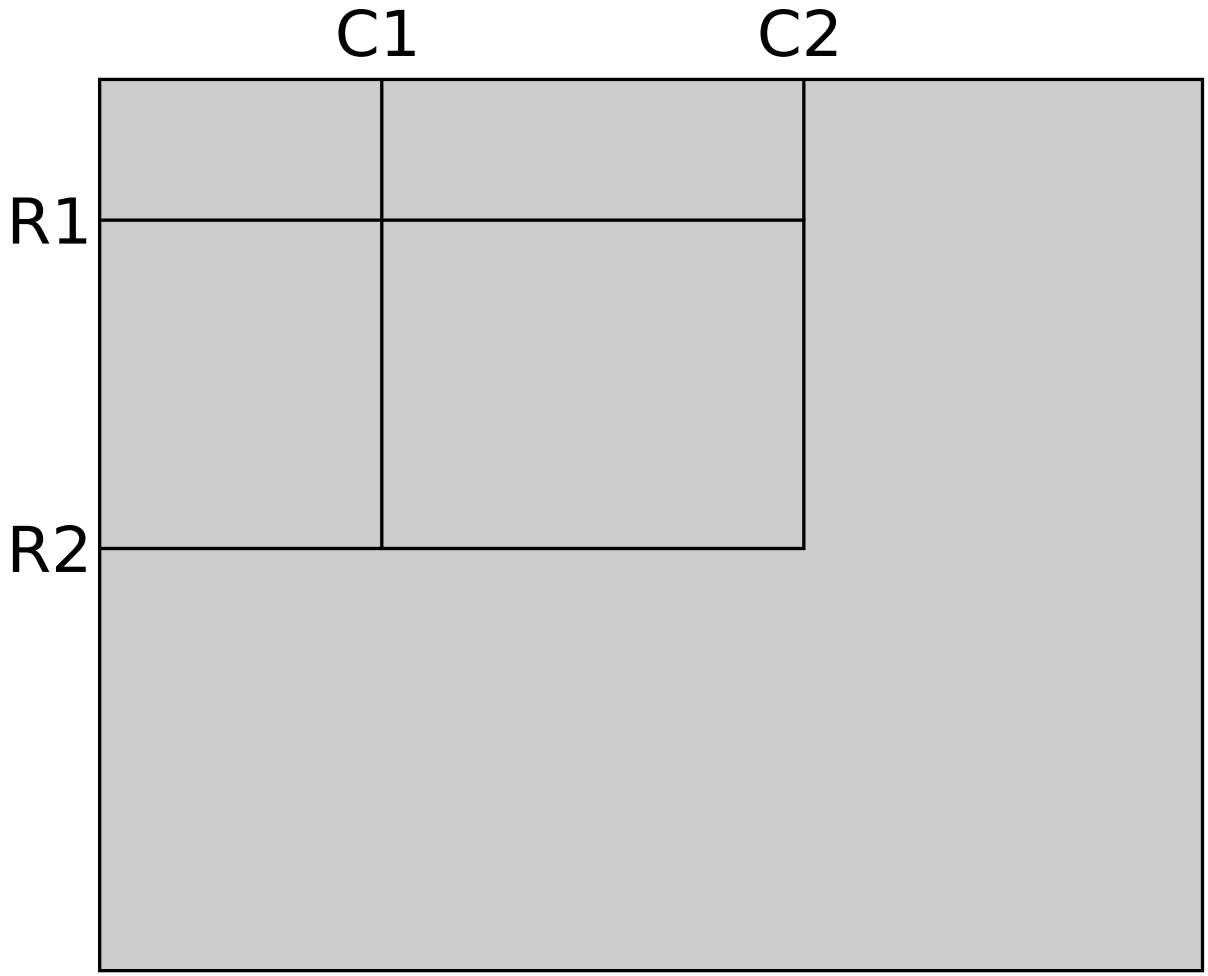
$$S[i,j] = \sum_{a=1}^{i} \sum_{b=1}^{j} A[a,b]$$

Per calcolare la somma da $[r_1, c_1]$ a $[r_2, c_2]$:

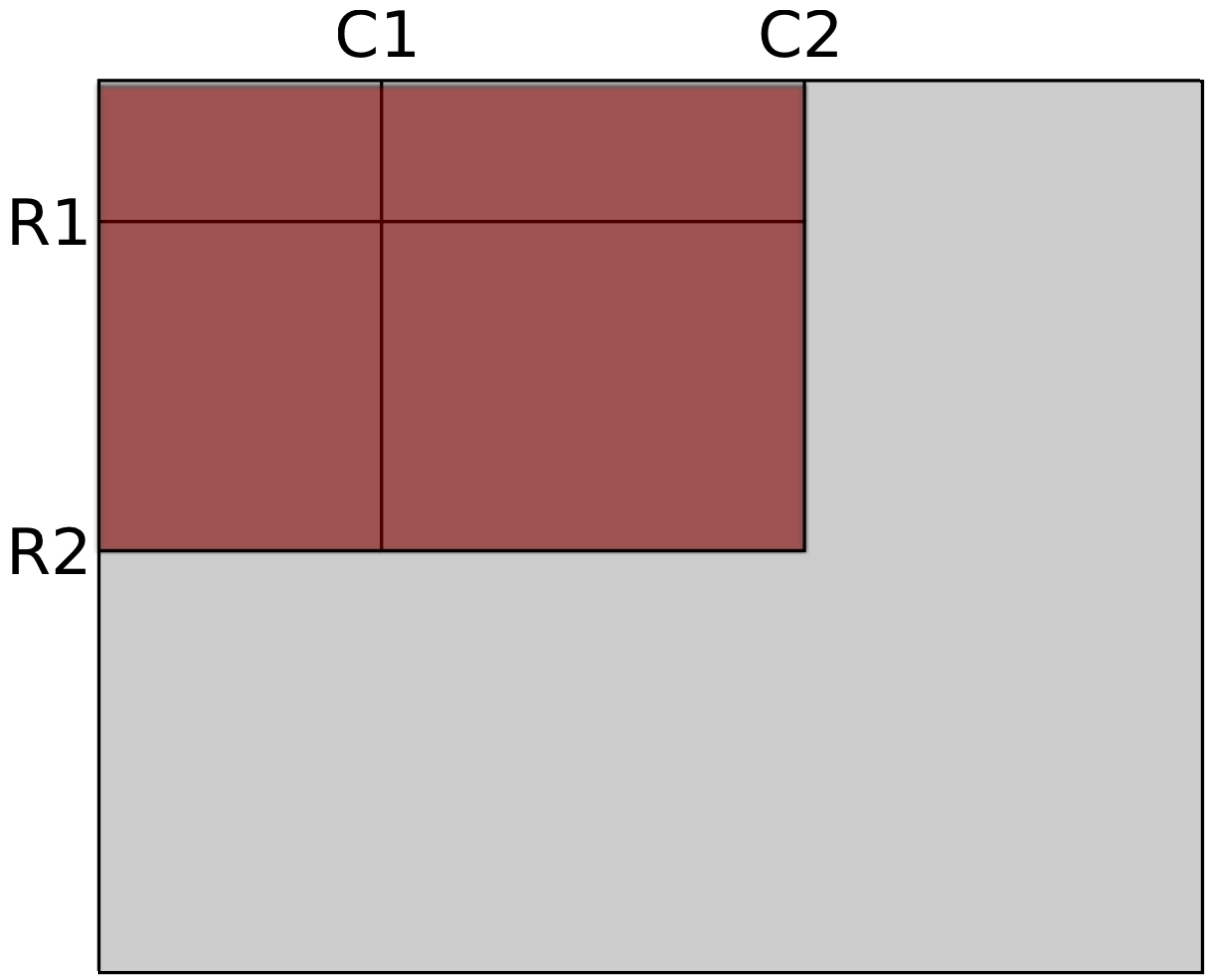
$$S[r_2, c_2] - S[r_2, c_1] - S[r_1, c_2] + S[r_1, c_1]$$

Sfruttando questa idea otteniamo un algoritmo $O((RC)^2)$

Soluzioni: Sottomatrice di somma massima

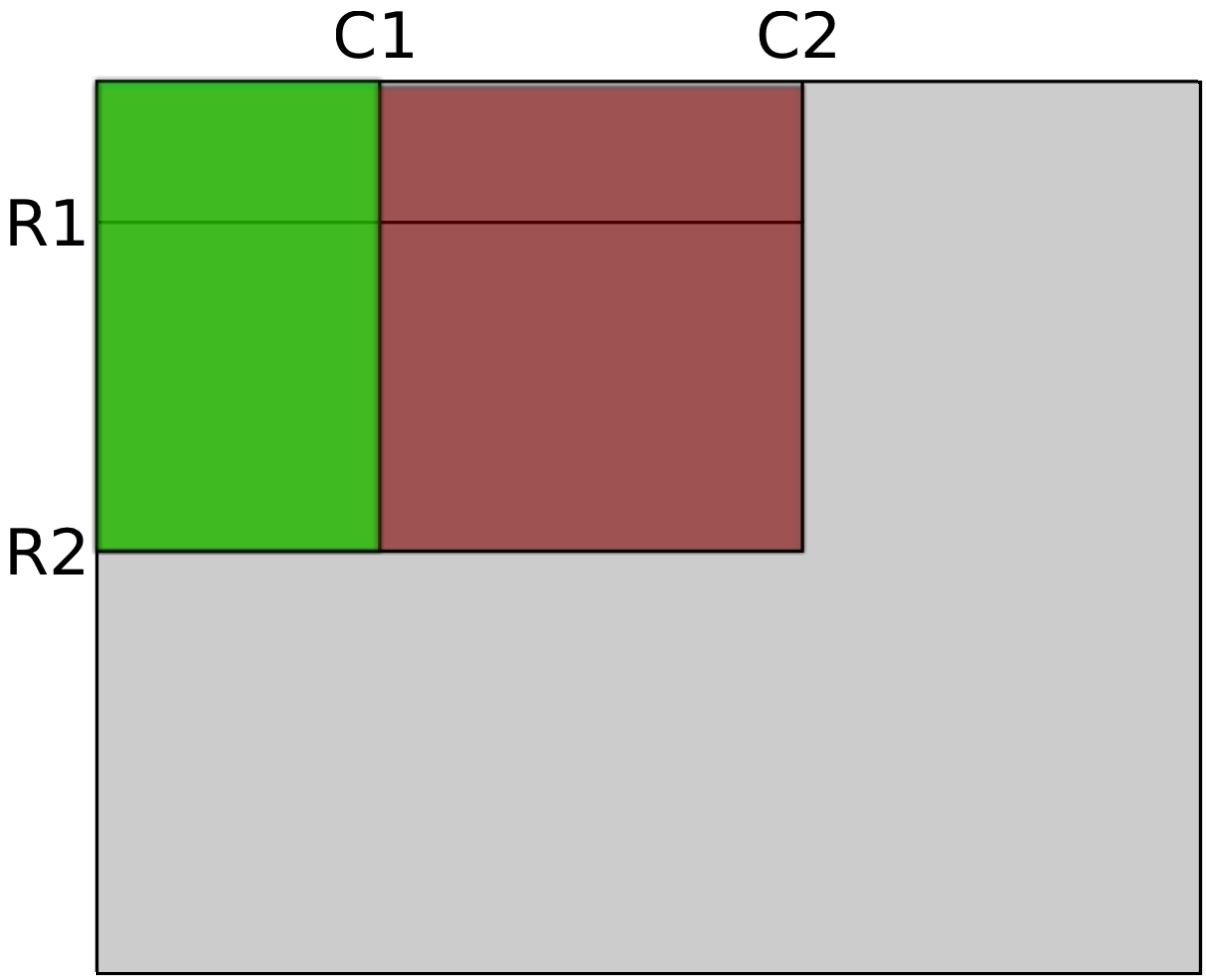


Soluzioni: Sottomatrice di somma massima



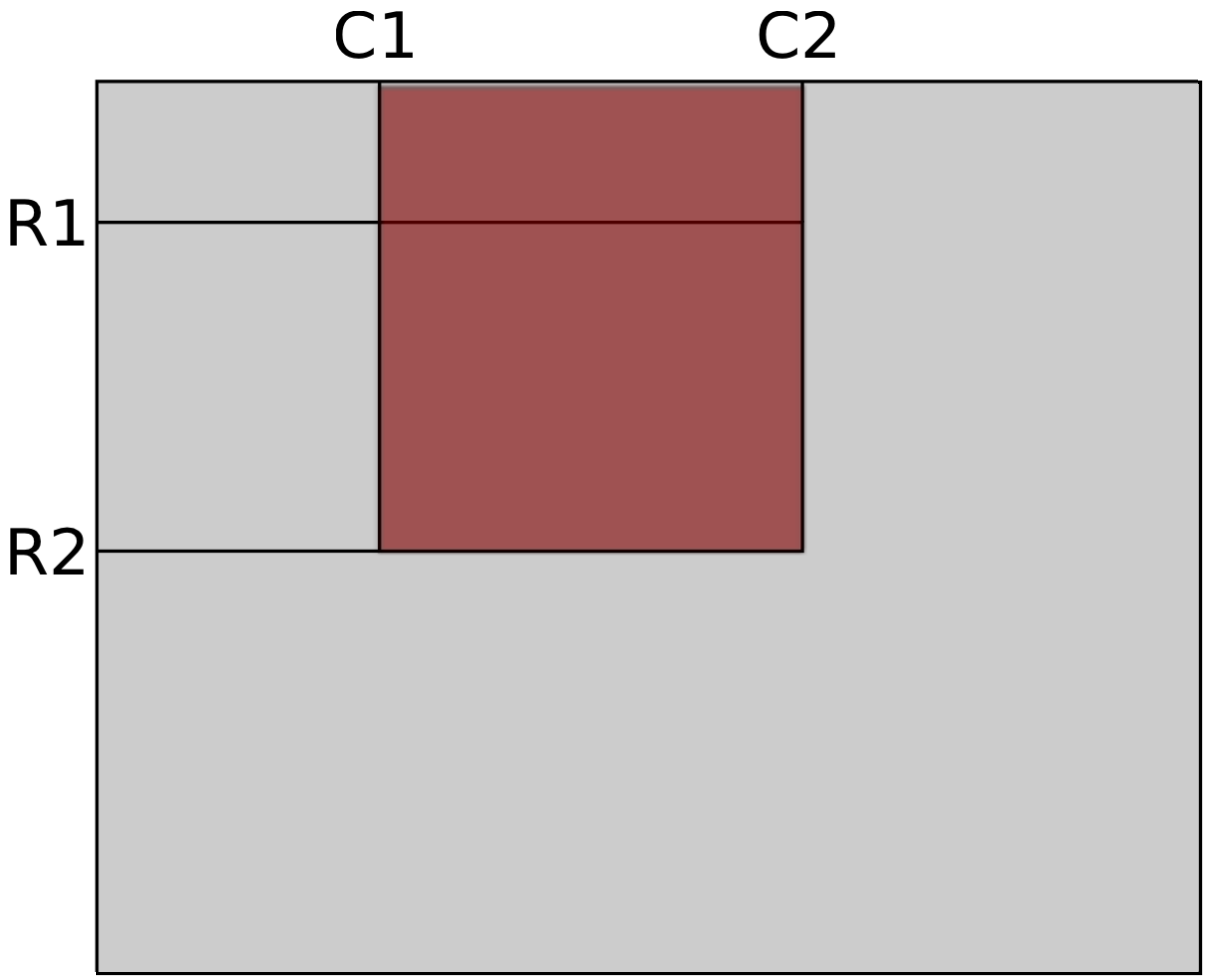
$S[r_2, c_2]$

Soluzioni: Sottomatrice di somma massima



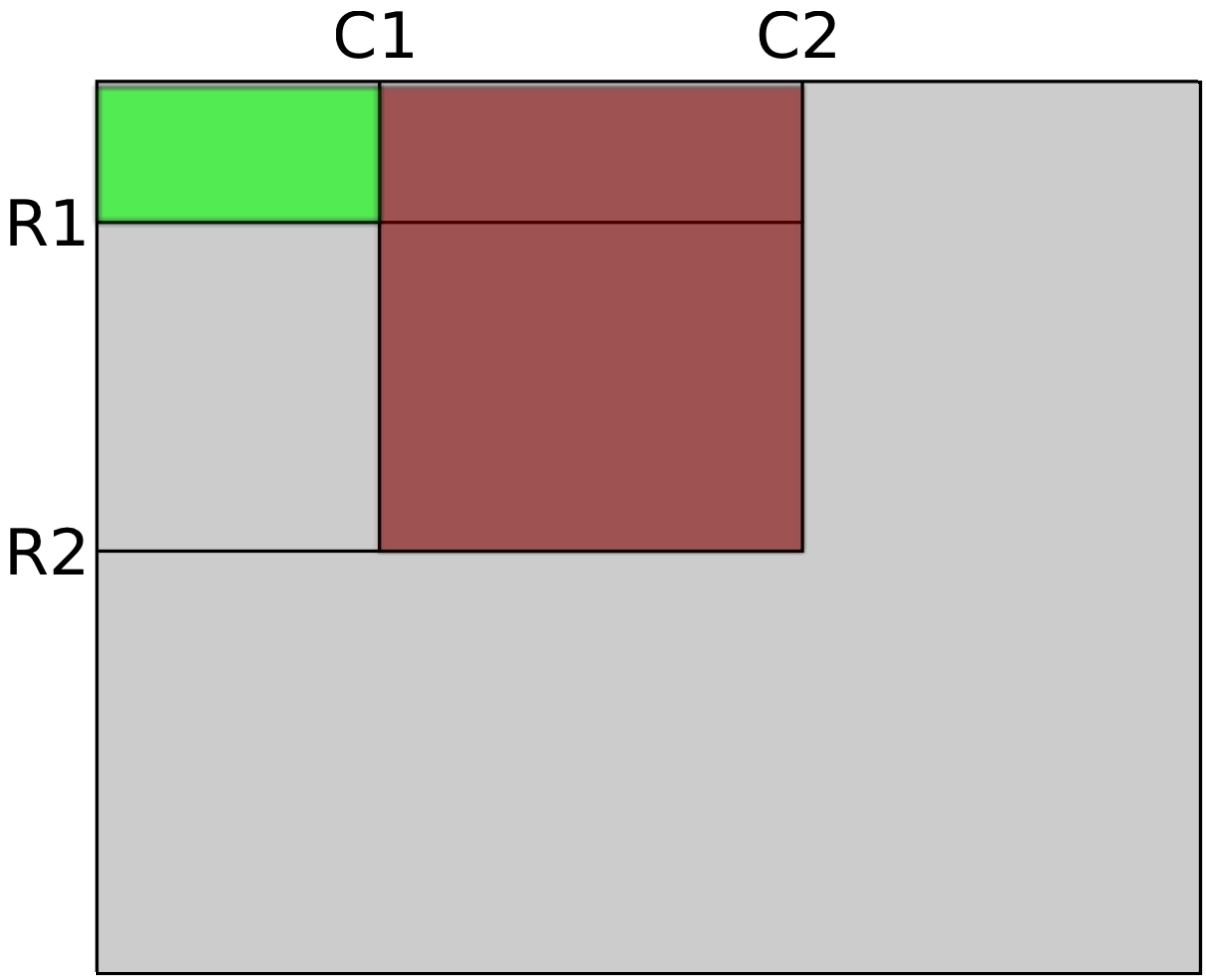
$$S[r_2, c_2] - S[r_2, c_1]$$

Soluzioni: Sottomatrice di somma massima



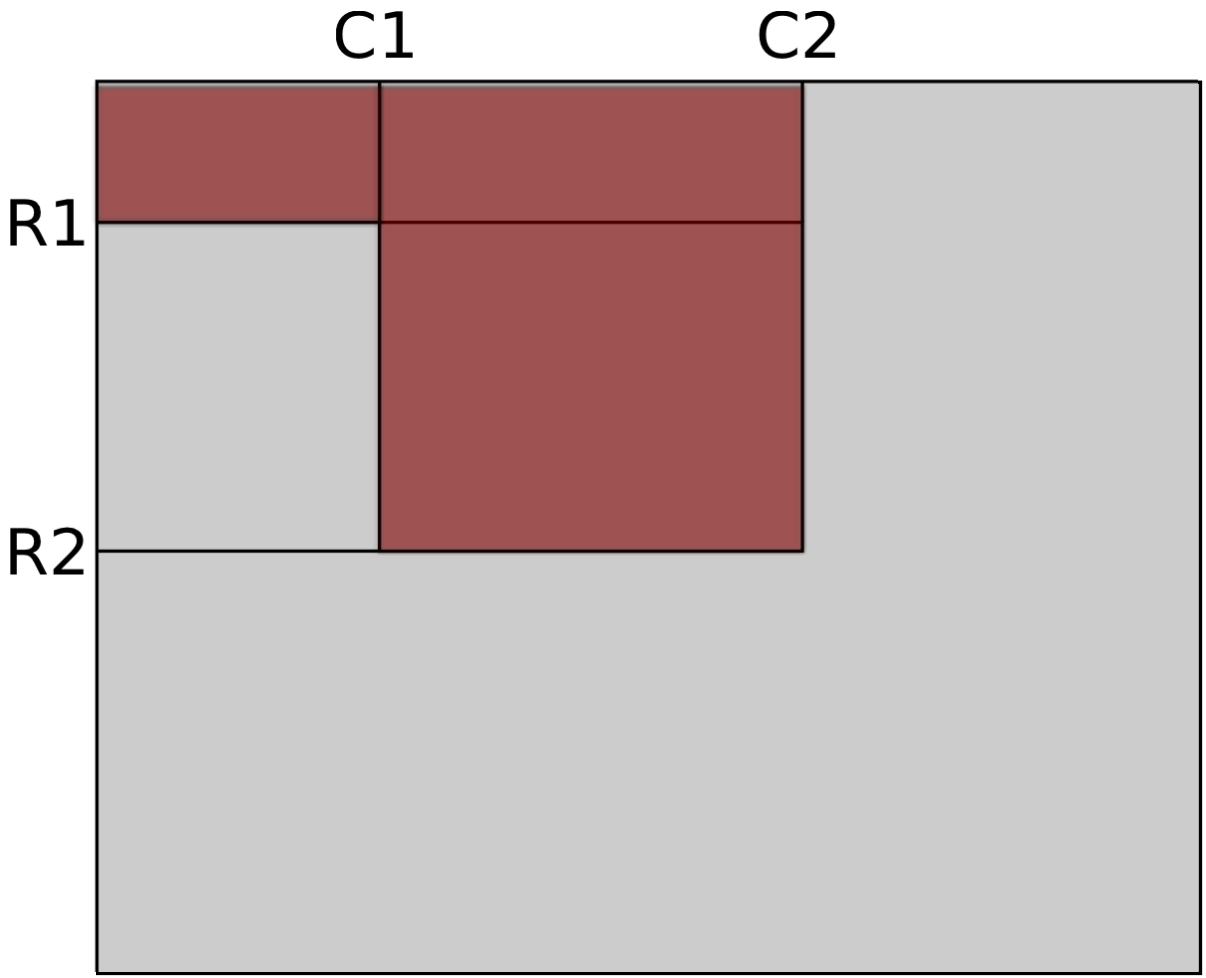
$$S[r_2, c_2] - S[r_2, c_1]$$

Soluzioni: Sottomatrice di somma massima



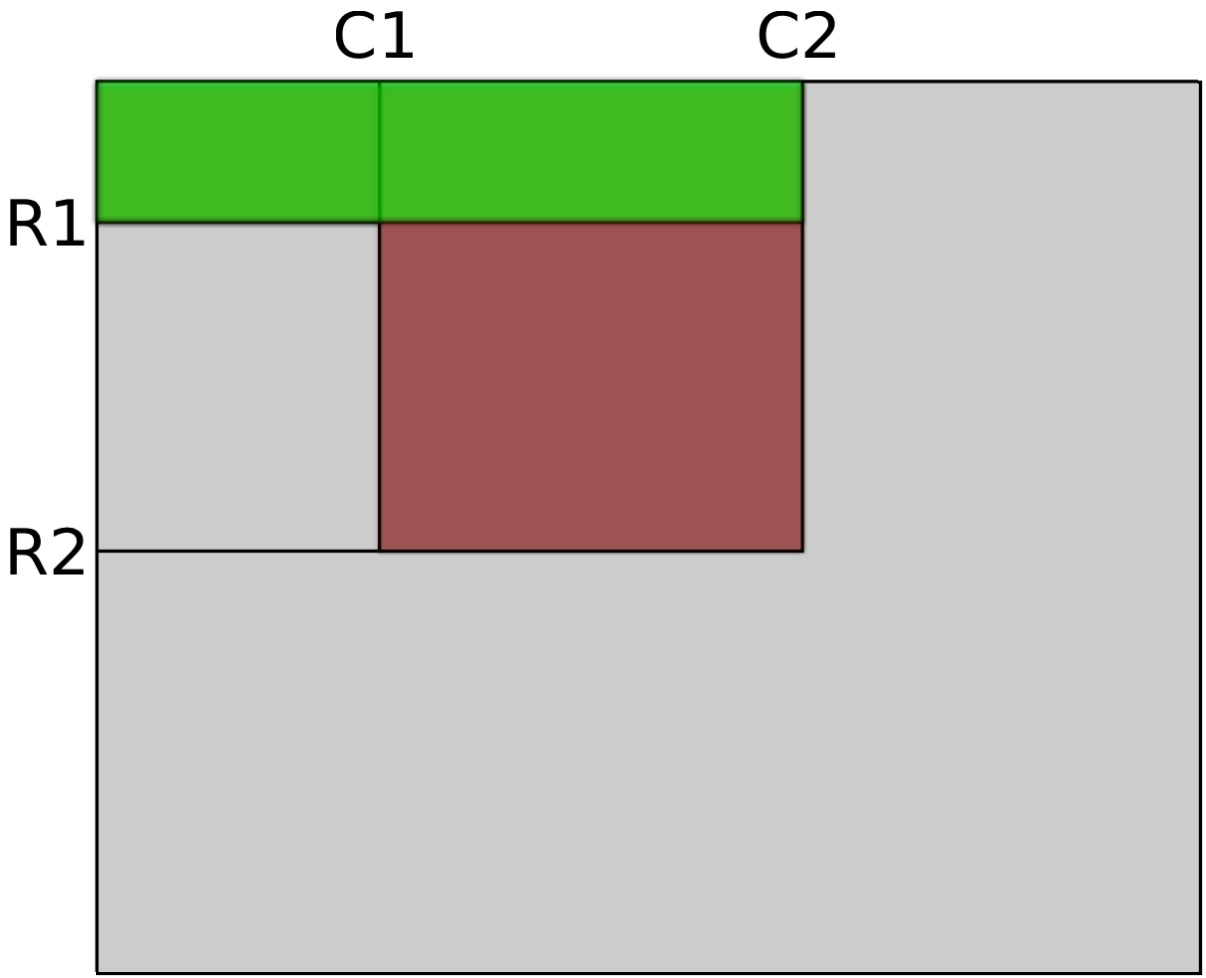
$$S[r_2, c_2] - S[r_2, c_1] + S[r_1, c_1]$$

Soluzioni: Sottomatrice di somma massima



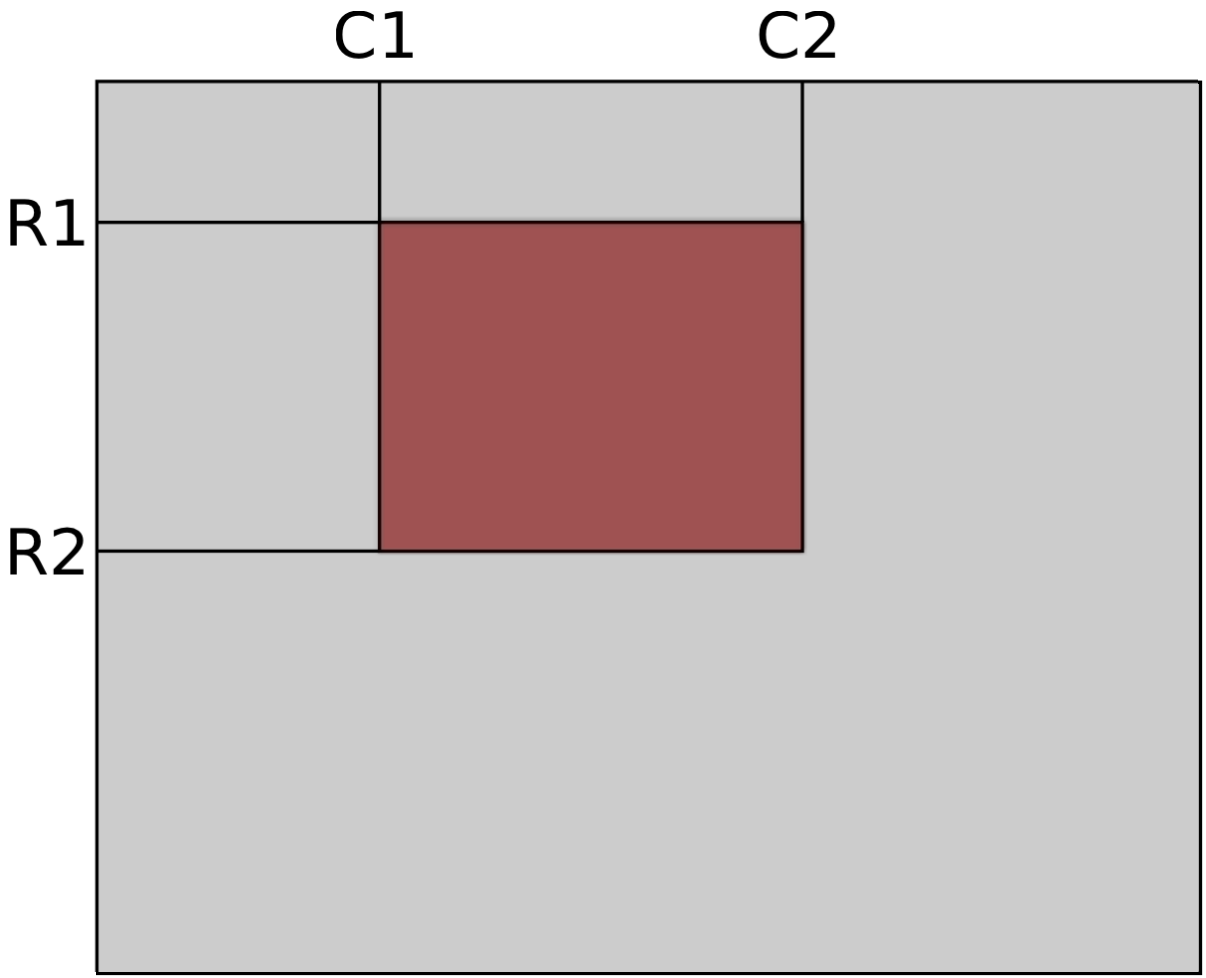
$$S[r_2, c_2] - S[r_2, c_1] + S[r_1, c_1]$$

Soluzioni: Sottomatrice di somma massima



$$S[r_2, c_2] - S[r_2, c_1] + S[r_1, c_1] - S[r_1, c_2]$$

Soluzioni: Sottomatrice di somma massima



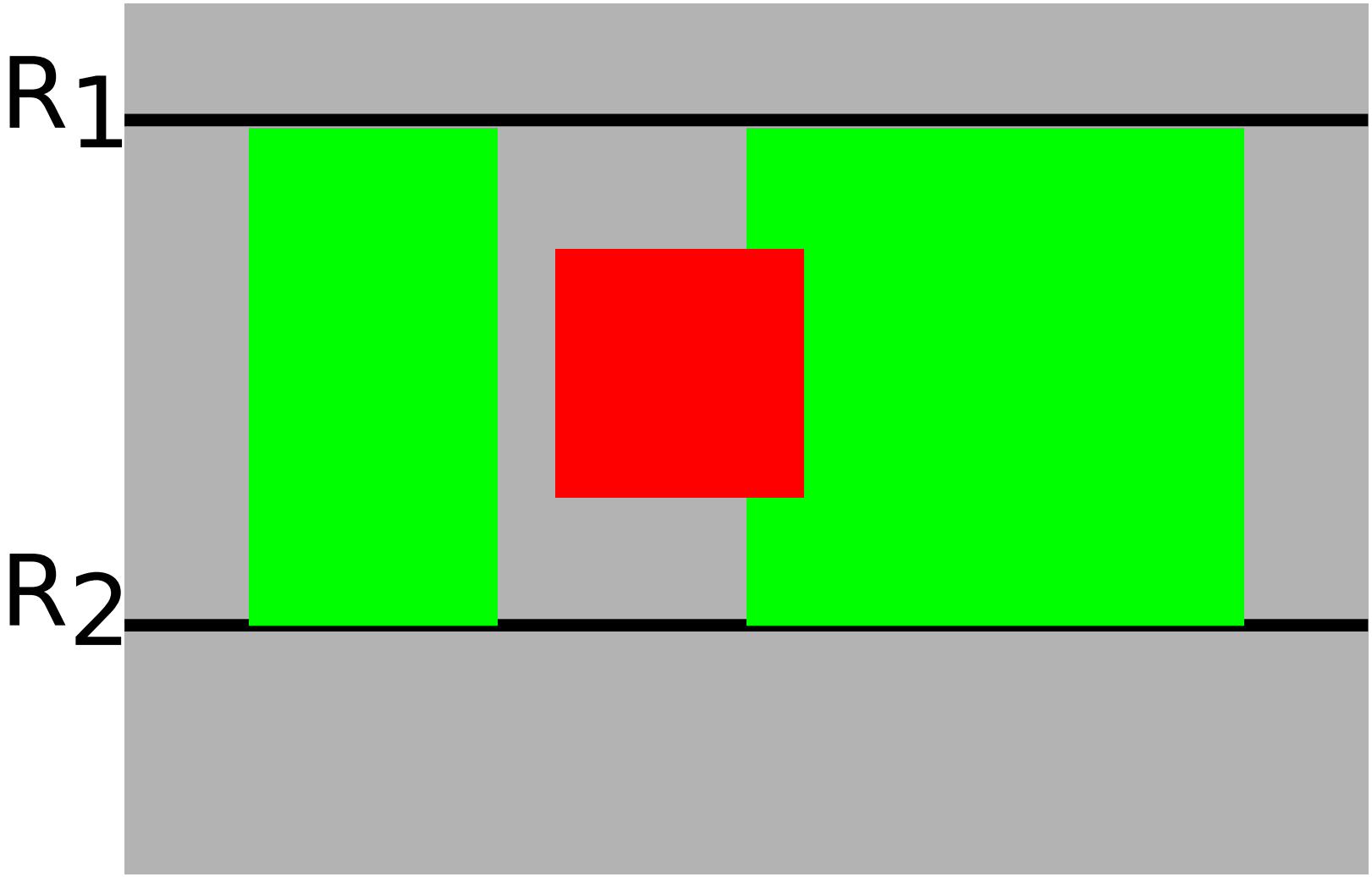
$$S[r_2, c_2] - S[r_2, c_1] + S[r_1, c_1] - S[r_1, c_2]$$

Soluzione ottima

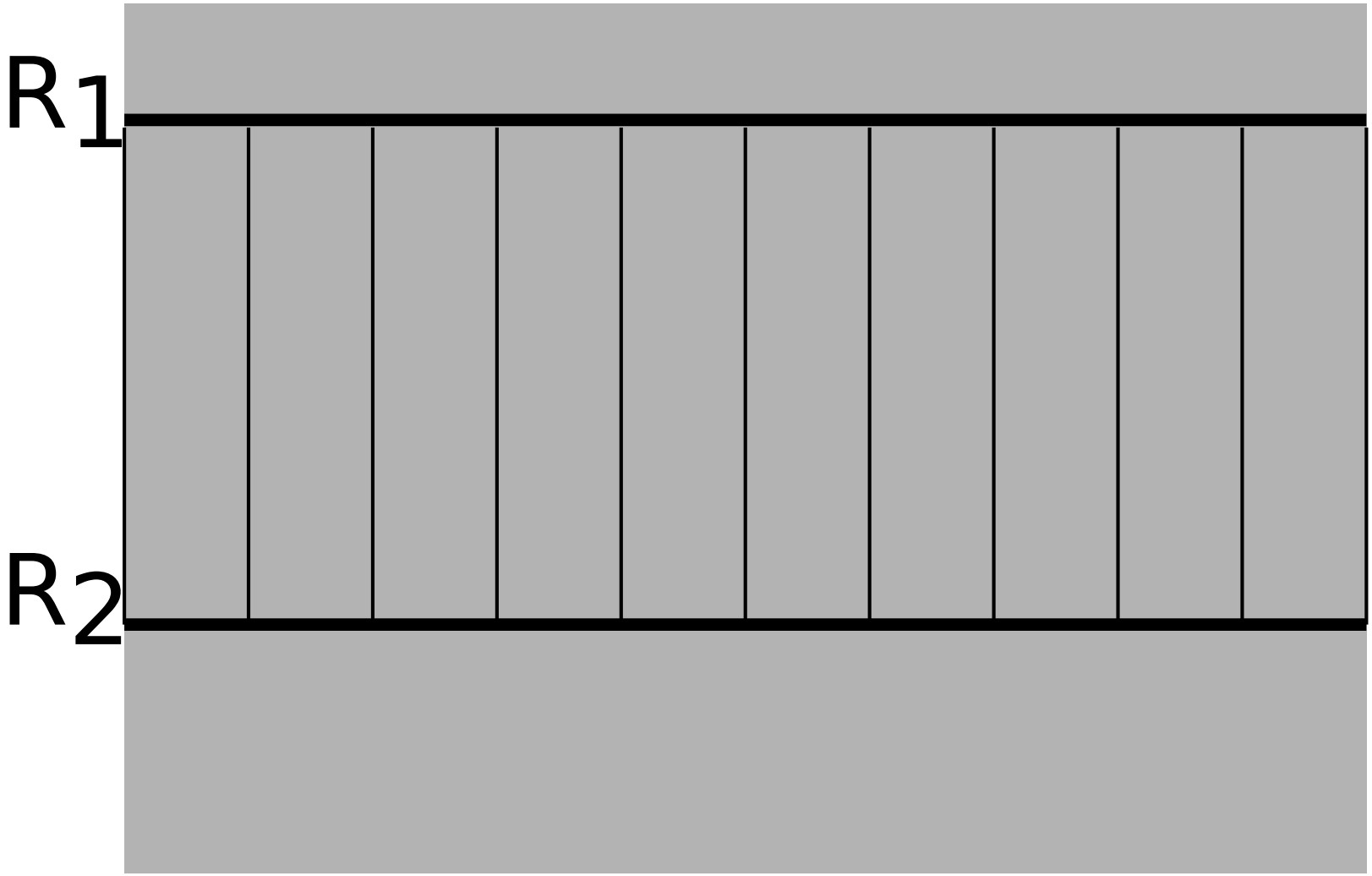
Vogliamo sfruttare la soluzione ottima $O(N)$ di sottosequenza per sottomatrice
Proviamo ad analizzare tutte le sottomatrici che partono dalla riga R_1 e finiscono alla riga R_2 (comprese)
Se $R_2 = R_1$ siamo nel caso della sottosequenza.
Negli altri casi?

Soluzione ottima

Soluzione ottima



Soluzione ottima



Se una sottomatrice contiene l'elemento $A[R_3][C']$ allora deve contenere tutti gli elementi su quella colonna che siano inclusi fra R_1 e R_2

Per ogni coppia R_1, R_2 creiamo un istanza del problema della sottosequenza di somma massima.

-1	2	4	3	2
1	3	-4	1	1
3	1	2	-5	2
2	-1	0	1	1
-2	4	2	-1	3

6	3	-2	-3	4

Esempio: $R_1 = 2, R_2 = 4$. La sottosequenza massima 6+3 corrisponde al rettangolo $(2, 1)(4, 2)$

Soluzione ottima

Per ogni R_1, R_2:

1. crea array $S[1..C]$
2. $S[i] = \sum_{j=R_1}^{R_2} A[j][i]$ (calcolato con somme parziali della colonna)
3. Usa l'algoritmo lineare per la sottosequenza di somma massima su S

Sorgenti disponibili sul sito.

CODING: STL

STANDARD TEMPLATE LIBRARY

```
1 #include <...>
2 using namespace std;
```

Documentazione online (anche su judge)

`http://www.cplusplus.com/reference/`

IFSTREAM E OFSTREAM

Lettura e scrittura su file. Come cout e cin, riconoscono il tipo delle variabili passate ed ignorano spazi ed invii.

```
#include <fstream>
using namespace std;

int main(){
  ifstream in("input.txt");
  int N;
  in>>N;
  for(int i=0;i<N;i++){
    int a;
    in>>a;
  }
}
```

```
#include<stdio.h>

int main(){
  FILE* f=fopen("input.
      txt","r");
  int N;
  fscanf(f,"%d",&N);
  for(int i=0;i<N;i++){
    int a;
    fscanf(f,"%d",&a);
  }
}
```

CODING: VECTOR

Equivalente all'arraylist di java.

```
1  #include<vector>
2  //Crea vector di interi
3  vector<int> intvec;
4  //Crea vector di 7 float inizializzati a 0.5
5  vector<float> floatvec(7,0.5);
6  //Accedi agli elementi
7  floatvec[2]=floatvec[5]+0.1;
8  //Aggiungi un elemento in fondo all'arraylist
9  intvec.push_back(231);
10 //Cicla sugli elementi:
11 for(int i=0;i<intvec.size();i++)
12   intvec[i]=12;
13 //Ridimensiona vector
14 intvec.resize(100);
```

Coding: Pair

Coppia di elementi.

```
1 #include <utility>
2 //pair di intero e float
3 pair<int,float> coppia1
4 //assegnazione elementi
5 coppia1.first=2;
6 coppia1.second=3.4;
7 coppia1=make_pair(15,0.4);
8 //coppia di coppie
9 pair<pair<int,int>, pair<int,int> > c;
```

CODING: SORT

```
1 #include <algorithm>
2 //ordinare un array di N elementi
3 sort(arr,arr+N);
4 //ordinare un vector
5 sort(vec.begin(),vec.end());
```

CODING: SORTING STRUCTS

```
1  #include <algorithm>
2  using namespace std;
3
4  struct stud{
5    int id;
6    int voto;
7  };
8
9  bool operator < (const stud a, const stud b){
10   return a.voto < b.voto;
11 }
12
13 int main(){
14   stud arr[2];
15   arr[0].id=1; arr[0].voto=30;
16   arr[1].id=2; arr[1].voto=20;
17   sort(arr,arr+2);
18 }
```

Coding: Coda

```
1  #include <queue>
2  //Dichiarare coda di interi
3  queue<int> q;
4  //Aggiungere un elemento alla coda
5  q.push(23);
6  //Leggere l'elemento in testa alla coda
7  int el=q.front();
8  //Eliminare l'elemento in testa alla coda
9  q.pop();
10 //Controllare se la coda e vuota
11 if(q.empty())
12     ...
```

Coding: Pila

```
1  #include <stack>
2  //Dichiarare pila di interi
3  stack<int> s;
4  //Aggiungere un elemento in cima alla pila
5  s.push(23);
6  //Leggere l'elemento in cima alla pila
7  int el=s.top();
8  //Eliminare l'elemento in cima alla pila
9  s.pop();
10 //Controllare se la pila e vuota
11 if(s.empty())
12     ...
```

Problemi

Testi completi su Judge.

Sorting

Implementate un algoritmo di ordinamento $N \log N$

Intervalli

Dato un insieme di intervalli temporali, scoprire il periodo più lungo non coperto da alcun intervallo.

Sorting pesato

Avete un array di interi. Ad ogni turno potete scambiare le posizioni di due interi, pagando la loro somma. Quale é il numero minimo di turni per ordinare l'array? Quanto é il prezzo minimo?