

# *ASD Laboratorio 01*

Cristian Consonni/Alessio Guerrieri

UniTN

10/10/2017

# CONTATTI

## ISTRUTTORI

Cristian Consonni ([cristian.consonni@unitn.it](mailto:cristian.consonni@unitn.it))

Alessio Guerrieri ([a.guerrieri@unitn.it](mailto:a.guerrieri@unitn.it))

## RICEVIMENTO

Consonni: via email e ufficio Open Space 9, Povo 2 (dopo il ponte, di fronte all'ufficio del prof. Montresor)

Guerrieri: via email

## SITI INTERNET

Slides laboratorio (caricate in giornata):

<http://judge.science.unitn.it/slides/>

**Judge:** <http://judge.science.unitn.it>

**Accesso a Judge tramite registrazione su:**

<http://judge.science.unitn.it/registration>

|       |              |
|-------|--------------|
| 10/10 | Introduzione |
| 31/10 | Ad-Hoc       |
| 14/11 | Grafi 1      |
| 28/11 | Grafi 2      |
| 05/12 | Progetto 1   |
| 12/12 | Progetto 1   |

## Progetto:

- 05 – 13 dicembre;
- Iscrizione dei gruppi ai progetti entro il **04 dicembre**:

<http://bit.ly/ASDprog>

# PERCHÉ FARE UN LABORATORIO

WHO CAN EXPLAIN  
frogSort ALGORITHM.



START WITH EMPTY LIST.  
FOR EACH INTEGER, PUT  
THAT NUMBER DEAD FLIES  
IN ONE BOX. THEN PUT FROG  
IN EACH BOX. WHEN FROG  
LEAVE BOX, APPEND THAT  
BOX'S FLY NUMBER  
TO LIST.



MORE FLY TAKE  
LONGER TO EAT.  
WHEN ALL FROGS  
GONE FROM BOXES,  
LIST ORDERED.



WHAT IS MAXIMUM  
STEP NUMBER?



$\log_{ing}(\text{boxes})$ .

VERY GOOD. NOW,  
HOMEWORK IS PROGRAM  
frogSort ON HOME FROGPUTER.



LATER...

BAH! ME UNDERSTAND  
BUT KEEP GETTING  
OFF-BY-FROG ERROR!



## 5 / 29

◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ ≡ ↺ 🔍 ↻

# OBIETTIVI DEL LABORATORIO

| CAPACITÀ                                            | ATTIVITÀ                                                |
|-----------------------------------------------------|---------------------------------------------------------|
| Sapere la differenza fra pseudocodice e chiacchiere | Passaggio da pseudocodice a codice                      |
| Utilizzare i concetti imparati a lezione            | Risoluzione di problemi                                 |
| Saper valutare l'efficienza di un algoritmo         | Test automatizzato usando dati di differenti dimensioni |

Useremo la Standard Template Library di C++ in modo da evitare la reimplementazione di strutture dati conosciute.

# NON OBIETTIVI

- Ottimizzazioni a basso livello

## SCRIVETE COSÌ

```
float f=...  
f*=pow(2,n);
```

## NON COSÌ

```
float f=...  
if (*(int*)&f & 0x7FFFFFFF) {  
    *(int*)&f += n << 23;  
}
```

*We should forget  
about small  
efficiencies, say  
about 97% of the  
time: premature  
optimization is the  
root of all evil*  
Donald Knuth

- Soluzioni lab precedente (con consegna sorgenti)
- Descrizione di 3/4 problemi:
  - ▶ Traduzione da pseudocodice a codice
  - ▶ Problema semplice
  - ▶ Problema complicato
  - ▶ Vecchio progetto (non tutte le settimane)
- Lavoro individuale/gruppo per resto laboratorio

Purtroppo, oggi ci saranno anche chiacchiere



Creato per l'edizione 2012 delle olimpiadi internazionali d'informatica

## FUNZIONAMENTO

- Per ogni problema il sistema ha dei file di input ed una soluzione “ufficiale”
- Le vostre soluzioni devono leggere i dati di input da “input.txt” e scrivono su “output.txt”
- Il sistema riceve il sorgente e lo esegue per ogni file di input con un time limit per il singolo caso
- La soluzione riceve un punteggio da 0 a 100, in base a quante volte ha scritto la risposta corretta in tempo

# ESEMPIO DI SOLUZIONE

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

int main() {
    int N,M;
    ifstream in("input.txt");
    in>>N>>M;
    ofstream out("output.txt");
    out<<N+M<<"\n";
    return 0;
}
```

# CMS: CONTEST MANAGEMENT SYSTEM

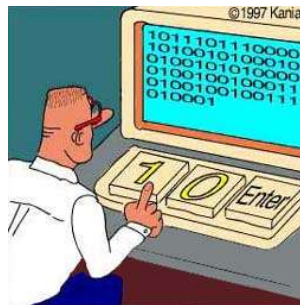
- Accessibile da `http://judge.science.unitn.it`
- Nome utente/password su:  
`http://judge.science.unitn.it/registration`
- Sorgenti in C/C++

## SISTEMA DI SVILUPPO

- (Emacs/vim/gedit) + terminale
- Netbeans + Plugin C/C++

Altre possibilità:

- Eclipse + Plugin C/C++
- Codeblocks
- Geany
- Atom
- ...



Real programmers code in binary.

- 1 progetto in questo semestre
- 2 progetti nel prossimo semestre con valutazione unificata
- Gruppi da 2 o 3 persone
- 8.5 giorni di tempo (~ 200 h)
- Sottoposizione usando CMS
- Progetto superato se la soluzione fa almeno 30 punti su 100
- Iscrizione su <http://bit.ly/ASDprog>



- È necessario superare almeno un progetto(\*) per accedere allo scritto
- I progetti completati durante il corso danno punti bonus allo scritto
- Primo progetto da 1 a 3 punti
- Secondo e terzo progetto da 1 a 2 punti
- Punteggio dato in maniera competitiva
- Il progetto non è una barriera aggiuntiva

(\*) per chi fa solo il primo modulo è necessario superare il primo

- Vietata collaborazione di alcun tipo fra i gruppi
- Potete chiedere agli assistenti in caso di difficoltà
- Abbiamo potenti mezzi...
- Copiando guadagnate al massimo 1/2 punti allo scritto
- Se vi becchiamo...



## NOTE DI COMPILAZIONE

- Sul server viene usato -DEVAL
- Consigliato C++ per le librerie
- Standard C++11 consigliato (più semplice!)

I miei esempi saranno C++11 (compila con `-std=c++0x`)

## STANDARD TEMPLATE LIBRARY

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

Documentazione online (anche su judge)

<http://www.cplusplus.com/reference/>

# IFSTREAM

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

## LETTURA INPUT

```
#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;
    }
```



# IFSTREAM

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

## SCRITTURA OUTPUT

...

```
ofstream out("output.txt");  
out<<N<<endl;  
for(int el:vec) {  
    out<<el<<endl;  
}  
  
return 0;  
}
```

# CODING: VECTOR

Equivalente all'arraylist di java.

```
#include<vector>
//Crea vector di interi
vector<int> intvec;
//Crea vector di 7 float inizializzati a 0.5
vector<float> floatvec(7,0.5);
//Accedi agli elementi
floatvec[2]=floatvec[5]+0.1;
//Aggiungi un elemento in fondo al vector
intvec.push_back(231);
//Cicla sugli elementi:
for(int i=0;i<intvec.size();i++)
    intvec[i]=12;
//Ridimensiona vector
intvec.resize(100);
```

## Coppia di elementi.

```
#include <utility>
//pair di intero e float
pair<int, float> coppial
//assegnazione elementi
coppial.first=2;
coppial.second=3.4;
coppial=make_pair(15, 0.4);
//coppia di coppie
pair<pair<int, int>, pair<int, int> > c;
```

# CODING: SORT

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

# CODING: SORTING STRUCTS

```
#include <algorithm>
#include <vector>
using namespace std;

struct stud{
    int id;
    int voto;
};

bool operator < (const stud a, const stud b){
    return a.voto < b.voto;
}

int main(){
    vector<stud> arr(2);
    arr[0].id=1; arr[0].voto=30;
    arr[1].id=2; arr[1].voto=20;
    sort(arr.begin(),arr.end());
}
```

```
#include <queue>
//Dichiarare coda di interi
queue<int> q;
//Aggiungere un elemento alla coda
q.push(23);
//Leggere l'elemento in testa alla coda
int el=q.front();
//Eliminare l'elemento in testa alla coda
q.pop();
//Controllare se la coda e vuota
if(q.empty())
    ...
```

```
#include <stack>
//Dichiarare pila di interi
stack<int> s;
//Aggiungere un elemento in cima alla pila
s.push(23);
//Leggere l'elemento in cima alla pila
int el=s.top();
//Eliminare l'elemento in cima alla pila
s.pop();
//Controllare se la pila e vuota
if(s.empty())
    ...
```

# NOTE SU C++11

- For-each
- auto
- Move operator

```
vector<int> arr= ...;  
for(int el:arr){  
    cout<<el<<endl;  
}  
for(int& el:arr){  
    el++;  
}  
auto d=23;  
for(auto& el:arr){  
    el+=d;  
}  
return arr;
```



# SOMMA DI DUE NUMERI

Dati due interi, sommateli.

INPUT.TXT

Due interi N,M separati da spazio

OUTPUT.TXT

Un intero, uguale alla somma di N e M.

Esempio:

`input.txt`

2 3

`output.txt`

5

# SOTTOSEQUENZA DI SOMMA MASSIMA

Data una sequenza di interi, trovare la sottosequenza di somma massima

## INPUT.TXT

N+1 righe: Il numero di elementi N sulla prima riga e gli N elementi nelle N righe seguenti.

Esempio:

input.txt

5  
3  
-2  
4  
1  
5

output.txt

11

# SOTTOMATRICE DI SOMMA MASSIMA

Data una matrice di interi, trovare la sottomatrice di somma massima

**INPUT.TXT**

R+1 righe: R e C (numero di righe e di colonne) sulla prima riga, C interi su ognuna delle seguenti R righe.

**Esempio:**

input.txt

```
3 4
2 -9 2 3
1 4 5 1
-2 3 4 1
```

output.txt

18

- Vecchio progetto di algoritmi
- Slides sul sito (secondo progetto, a. a. 2014/2015):  
`http://judge.science.unitn.it/slides/asd14b/prog2.pdf`
- Esiste soluzione con Programmazione Dinamica
- Esiste anche soluzione ad-hoc.

# LAVORATE!

- ❶ Se non avete già un account:  
`http://judge.science.unitn.it/registration`
- ❷ Implementate una soluzione per il problema della somma e testatela su  
`http://judge.science.unitn.it`
- ❸ Risolvete uno (o entrambi) gli altri problemi
- ❹ Non usate judge come compilatore!
- ❺ Studenti di matematica mi vengano a parlare

## NOTE

- I file C++ devono avere l'estensione `.cpp`