

Algorithm Unibo

Introduzione al progetto

Giovanni Spadaccini, Angelo Huang, Samuele Marro

Università di Bologna

t.me/algo_unibo

Table of Contents

- 1 Introduzione al progetto
- 2 Competitive Programming
- 3 Machine Learning
- 4 Laboratorio CP

Introduzione

Abbiamo attualmente questi gruppi di studenti in campo informatico:

- ULISSE (Cybersecurity)
- RaspiBo (Hardware, Raspberry e simili)

- Obiettivo: Fare un gruppo studentesco sugli **algoritmi**
 - Algoritmi dal punto di vista del Competitive Programming
 - Algoritmi di Machine Learning
 - Applicazione algoritmi studiati nel corso di ASD
 - Sincronizzazione di reti
 - problemi di CP
 - E molto altro!
- Focus sull'aspetto pratico

Obiettivi:

- 1 creare un ambiente favorevole allo studio e approfondimento di argomenti in campo informatico, con un focus sugli **algoritmi**
- 2 affiancare allo studio prettamente teorico degli argomenti, uno studio a gruppi basato sulla *pratica* e la *collaborazione fra pari*.

Valori:

-
-

Obiettivi:

- 1 creare un ambiente favorevole allo studio e approfondimento di argomenti in campo informatico, con un focus sugli **algoritmi**
- 2 affiancare allo studio prettamente teorico degli argomenti, uno studio a gruppi basato sulla *pratica* e la *collaborazione fra pari*.

Valori:

-
-

Obiettivi:

- 1 creare un ambiente favorevole allo studio e approfondimento di argomenti in campo informatico, con un focus sugli **algoritmi**
- 2 affiancare allo studio prettamente teorico degli argomenti, uno studio a gruppi basato sulla *pratica* e la *collaborazione fra pari*.

Valori:

- **Condivisione di conoscenze**

Obiettivi:

- 1 creare un ambiente favorevole allo studio e approfondimento di argomenti in campo informatico, con un focus sugli **algoritmi**
- 2 affiancare allo studio prettamente teorico degli argomenti, uno studio a gruppi basato sulla *pratica* e la *collaborazione fra pari*.

Valori:

- **Condivisione di conoscenze**
- **Pratico nel teorico**

Informazioni logistiche

- Incontri settimanali
- Prossimi probabilmente a Ranzani, ancora da vedere.

Tipologie di incontri

- Laboratori: niente presentazioni, esercizi in gruppo e discussioni su temi di interesse
- Seminari: presentazione su un particolare tema (un problema, un algoritmo, una discussione)
- Mix!

Informazioni logistiche

- Incontri settimanali
- Prossimi probabilmente a Ranzani, ancora da vedere.

Tipologie di incontri

- Laboratori: niente presentazioni, esercizi in gruppo e discussioni su temi di interesse
- Seminari: presentazione su un particolare tema (un problema, un algoritmo, una discussione)
- Mix!

Competitive Programming

*A programming competition generally involves the host presenting a set of **logical or mathematical problems**, to the contestants. Contestants are required to write **computer programs** capable of solving these problems. Judging is based mostly upon number of problems solved (Wikipedia)*

Sono dei *servizi web* in cui si possono trovare esercizi e sottoporre soluzioni ai problemi

- Codeforces (general) - <https://codeforces.com/>
- CSES (raccolta di problemi classici) - <https://cses.fi/problemset/>
- Olinfo (sito olimpiadi di Informatica italiane) - <https://training.olinfo.it/>

- Southwestern Europe Regional Contest
 - Unibo partecipa da 5 anni
 - È un bel modo per conoscere aziende dove andare per fare intership/tirocinio
- Itacpc
 - Gara tra le università italiane
 - L'anno scorso una squadra di bologna è arrivata prima

Perché fare CP?

- 1 Scrittura e comprensione di algoritmi

Perché fare CP?

1 Scrittura e comprensione di algoritmi

```
#include <bits/stdc++.h>
using namespace std;
#define N 805
#define ll long long
#define pb push_back
const ll INF=1e18;
int T,n,m,c,a[N],ord[N];ll dst[N],e[N][N];
bool cmp(int x,int y) {return a[x]<a[y];}
struct Node {ll x,y;bool operator < (Node t) const {return x==t.x?y>t.y:x<t.x;}}dp[N];
void slv()
{
    scanf("%d %d %d",&n,&m,&c);
    for(int i=1;i<=n;++i) ord[i]=i,dp[i]=(Node) {INF,INF},scanf("%d",&a[i]);
    for(int i=1;i<=n;++i) for(int j=1;j<=n;++j) e[i][j]=INF;
    sort(ord+1,ord+n,cmp);
    for(int i=1,u,v,w;i<=m;++i) scanf("%d %d %d",&u,&v,&w),e[u][v]=min(e[u][v],0ll+w);
    for(int k=1;k<=n;++k) for(int i=1;i<=n;++i) for(int j=1;j<=n;++j)
        e[i][j]=min(e[i][j],e[i][k]+e[k][j]);dp[1]=(Node) {0,c};
    for(int i1=1,i=ord[i1];i1<=n;++i1,i=ord[i1])
        for(int j1=1,j=ord[j1];j1<=i1;++j1,j=ord[j1]) if(e[j][i1]<INF)
        {
            ll w=e[j][i1],t1;Node t=dp[j];
            if(w<t.y) t.y=w;else w=t.y,t1=(w-1)/a[j]+1,t.x+=t1,t.y=t1*a[j]-w;
            dp[i1]=min(dp[i1],t);
        }printf("%lld\n",dp[n].x<INF?dp[n].x:-1);
}
int main()
{
    scanf("%d",&T);
    while(T--) slv();return 0;
}
```

Perché fare CP?

- 1 Scrittura e comprensione di algoritmi
- 2 Debugging, testing, e individuazione di errori logici.

Perché fare CP?

- 1 Scrittura e comprensione di algoritmi
- 2 Debugging, testing, e individuazione di errori logici.
- 3 Interview in big tech (Google, Amazon, Meta etc...)

- <https://neetcode.io/>

Struttura di un problema di CP

Time limit: 1.00 s **Memory limit:** 512 MB

You are given a list of n integers, and your task is to calculate the number of *distinct* values in the list.

Input

The first input line has an integer n : the number of values.

The second line has n integers x_1, x_2, \dots, x_n .

Output

Print one integer: the number of distinct values.

Constraints

- $1 \leq n \leq 2 \cdot 10^5$
- $1 \leq x_i \leq 10^9$

Example

Input:

```
5
2 3 2 2 3
```

Output:

```
2
```

Esempio tratto da <https://cses.fi/problemset/task/1621>

Struttura di un problema di CP

- 1 Descrizione del problema (c'è anche una storia di solito :D)
- 2 Formato dell'input
- 3 Formato dell'output
- 4 Constraints del problema (utile per fare una stima del costo dell'algoritmo finale)
- 5 Esempio di testcase (input e output) (utile per testare localmente)

Linguaggi più comuni

- **C++** è di gran lunga il linguaggio di riferimento per CP
 - ① velocità di esecuzione
 - ② grande versatilità delle classi e funzioni della STL (Standard Template Library)
 - ③ conciso, permette di scrivere algoritmi con garanzie dei tipi e senza boilerplate eccessivi.
- Molti altri linguaggi sono accettati dai Judge (Java, Kotlin Python etc..., a seconda del judge)

Documentazione libreria standard [C++ Reference](#)

Allowed Programming Languages
You can restrict programming languages participants can use to submit solutions. We recommend that you use this feature only if there are strong reasons to limit the languages of the contest.

☐ Allow all supported languages (recommended)

Language Groups

Name	Languages	
C	GNU C11	<input type="checkbox"/>
C#	Mono C#	<input type="checkbox"/>
C++	Clang++17 Diagnostics GNU C++11 GNU C++14 GNU C++17 MS C++ MS C++ 2017	<input checked="" type="checkbox"/>
D	D	<input type="checkbox"/>
Go	Go	<input type="checkbox"/>
Haskell	Haskell	<input type="checkbox"/>
Java	Java 11 Java 8	<input checked="" type="checkbox"/>
JavaScript	JavaScript Node.js	<input type="checkbox"/>
Kotlin	Kotlin	<input type="checkbox"/>
Ocaml	Ocaml	<input type="checkbox"/>
Pascal	Delphi FPC PascalABC.NET	<input type="checkbox"/>
Perl	Perl	<input type="checkbox"/>
PHP	PHP	<input type="checkbox"/>
Python	Python 2 Python 3 PyPy 2 PyPy 3	<input checked="" type="checkbox"/>
Ruby	Ruby	<input type="checkbox"/>

Esempi di linguaggi accettati su codeforces

Template e Tools

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

```
using namespace std;
```

```
int main() {}
```


Seminario: Machine Learning

- Samuele Marro, studente magistrale ad Artificial Intelligence
- Tutor del corso "Introduzione all'Apprendimento Automatico"
- Membro del Collegio Superiore, area STEM
- Ricercatore con il DISI
 - Attacchi avversariali (come ingannare un'AI)
 - Hyperparameter tuning (come addestrare bene un'AI)
 - AI per social engineering (come usare un'AI per ingannare gli umani)

- 1959: Rosenblatt inventa il Percettrone
- Anni '80: Il Multi-Layer Perceptron (la prima "rete neurale") acquisisce popolarità
- 2012: AlexNet vince l'ImageNet Large Scale Visual Recognition Challenge. Inizio simbolico del Deep Learning
- 2022: Un'opera di Midjourney vince il primo premio alla gara di arte della Colorado State Fair
- Marzo 2023: GPT-4 passa a pieni voti l'esame per diventare avvocato negli USA

Non è cambiato nulla!

- Alla fine un algoritmo di ML prende una rete neurale "non addestrata" e le insegna ad approssimare una funzione
 - $y = x^2$
 - "Data un'immagine, restituisci 1 se è l'immagine di un gatto e 0 altrimenti"
 - "Dato il nome di un autore, scrivi un nuovo libro nel suo stile"
 - "Data una domanda, restituisci la risposta che darebbe un essere umano"
- In 40 anni, il grosso di ricerca è stato su come creare reti sempre più grandi...
- ...ma gli algoritmi e i principi sono gli stessi
- Obiettivo: insegnarvi in 2 ore a creare una rete neurale e addestrarla

- La teoria del ML si basa su Algebra Lineare (60%), Statistica (20%) e Calcolo Numerico (20%)...
- ...ma le librerie semplificano tutto
- Prerequisito minimo: sapere cos'è un vettore
- Se sapete usare NumPy, avete già capito il 90%
- Linguaggio: Python
- Libreria: Keras

Laboratorio CP

Collecting Numbers

Dato un array che contiene ogni numero compreso tra 1 e n esattamente una volta. Il tuo compito è raccogliere i numeri da 1 a n in ordine crescente.

Ad ogni round, si passa attraverso l'array da sinistra a destra e si raccolgono quanti più numeri possibile. Quale sarà il numero totale di round?

Tratto da: - <https://cses.fi/problemset/task/2216>

Soluzione n^2

```
int solve_n2(vector<int> n) {  
    int next = 1;  
    int sol = 1;  
    int current = 0;  
    while (true) {  
        if (n[current] == next) {  
            next++;  
        }  
        if (next == n.size() + 1)  
            break;  
        current++;  
        if (current >= n.size()) {  
            sol++;  
            current = 0;  
        }  
    }  
}
```


Soluzione $n \log n$

```
int solve_nlogn(vector<int> n) {  
    vector<array<int, 2>> a;  
    for (int i = 0; i < n.size(); i++) {  
        a.push_back({n[i], i});  
    }  
    sort(a.begin(), a.end());  
    int sol = 1;  
    for (int i = 1; i < n.size(); i++) {  
        if (a[i][1] < a[i - 1][1]) {  
            sol++;  
        }  
    }  
    return sol;  
}
```

Soluzione n

```
int solve_n(vector<int> n) {  
    vector<bool> k(n.size() + 1, false);  
    k[0] = true;  
    int sol = 1;  
    for (int i = 0; i < n.size(); i++) {  
        k[n[i]] = true;  
        if (!k[n[i] - 1]) {  
            sol++;  
        }  
    }  
    return sol;  
}
```

Numeri Distinti

Data una lista di n numeri in input, restituire il numero di elementi distinti nella lista

Provate a scrivere il codice c++ per risolvere questo problema ed eseguitelo sul sito

<https://cses.fi/problemset/task/1621>

- Subarray Sum II: <https://cses.fi/problemset/task/1661>
- Sum of Two Values: <https://cses.fi/problemset/task/1640>
- Reading Book: <https://cses.fi/problemset/task/1631>
- Traffic Lights <https://cses.fi/problemset/result/290742/>

Altri da Codeforces:

<https://codeforces.com/problemset/problem/1749/B>

<https://codeforces.com/contest/1742/problem/D>

- Vettori Dinamici: `vector<tipo>`
- Sort: `sort (container.begin(), container.end())`
- Set: `set<tipo>`
- Map: `map<tipoChiave, tipoValore>`
- Input: `cin >> variabile;`
- Output: `cout << variabile`