



**Università degli Studi di Udine**

Dipartimento di Scienze Matematiche, Informatiche e Fisiche  
Nucleo di Ricerca in Didattica dell'Informatica



**Piano Lauree Scientifiche**

Scienze e Tecnologie Informatiche

## **COMPRESSIONE E COMPRESIONE**

### **Un'escursione algoritmica fra informazioni e bit**

#### **Campus telematico per studentesse e studenti della scuola secondaria di secondo grado**

**14–18 Giugno 2021, ore 9:30 – 17:30**

Canali della piattaforma Microsoft Teams  
dell'Università di Udine

L'iniziativa, inserita nel Progetto Nazionale Lauree Scientifiche per l'Informatica e fruibile nell'ambito dei Percorsi per le Competenze Trasversali e per l'Orientamento (PCTO), è rivolta a studenti e studentesse della scuola secondaria di secondo grado interessati/e alla programmazione e ad approfondire la prospettiva algoritmica. Come prerequisiti per una proficua partecipazione al Campus, oltre alla motivazione a svolgere le attività menzionate, è necessario conoscere le basi della programmazione strutturata, nonché essere in grado di sviluppare semplici programmi codificandoli in un (qualsiasi) linguaggio di programmazione.

Le attività si svolgeranno a distanza, per via telematica attraverso i canali UniUD della piattaforma Microsoft Teams, orientativamente dalle 9:30 alle 12:30 e dalle 14:30 alle 17:30. Saranno inoltre assegnati dei compiti che i partecipanti affronteranno autonomamente, individualmente o in piccoli gruppi, dopo la conclusione delle sessioni comuni.

#### **Temi del campus**

Le attività pratiche, progettuali e di sviluppo di programmi, avranno come filo conduttore un problema di un certo interesse pratico: la compressione (e ripristino) di documenti attraverso la codifica di Huffman. Oltre al persistente interesse applicativo della tecnica *lossless* (senza perdita di informazione) introdotta da David Huffman nel 1952, le cui proprietà sono ottimali sotto certe ipotesi, lo sviluppo di simili strumenti costituisce una stimolante palestra per esplorare una varietà di strutture dati e soluzioni algoritmiche. Si tratta comunque dello sviluppo di strumenti alla portata di allieve e allievi che rispondono ai prerequisiti indicati, e ci aspettiamo che a conclusione del campus i partecipanti riusciranno a raggiungere la piena padronanza dei programmi che li realizzano.

Oltre alle attività progettuali e pratiche a cui si è accennato, saranno proposte delle brevi conferenze su tematiche collegate (come il confronto con tecniche di compressione *lossless* che possono raggiungere fattori di compressione ancora più elevati e/o con altre tecniche *lossy*, cioè con perdita parziale di informazione, come quelle utilizzate in ambito multimediale per comprimere le rappresentazioni di immagini e suoni).

Le note riportate fin qui rendono conto della prima parola chiave nel titolo: *compressione*. Quanto alla seconda e più importante parola chiave, la *comprensione* si presta ad essere letta in questo contesto secondo tre accezioni diverse:

1. Tecniche di compressione come quella basata sulla codifica di Huffman prescindono completamente dalla *comprensione* del contenuto del documento, cioè al fine di poterle applicare è irrilevante essere in grado di interpretare il significato dei simboli che costituiscono il documento originale — conta solo come i dati sono rappresentati, e questi vengono trattati in termini di mere sequenze di bit.

2. Benché l'operazione di compressione sia indipendente da quello che il documento può significare, una tecnica *lossless* non comporta perdita di informazione, e quindi nulla di quanto può contribuire alla *comprensione* del documento originale va perduto nella versione compressa, da cui l'originale può essere ripristinato integralmente. (Ci possono essere invece implicazioni anche riguardo la comprensione se si fa uso di tecniche di compressione *lossy*.)
3. Infine, la *comprensione* è pienamente in gioco, ma su un piano diverso, quando si voglia dominare un processo algoritmico, pensandolo sempre in relazione al problema che si sta affrontando. È questa chiaramente l'accezione che più conta ai fini degli obiettivi del campus.

### Obiettivi del campus

I principali obiettivi delle attività sono:

- affinare la capacità di pensare in termini algoritmici, prefigurando diverse strategie risolutive, confrontandole e valutandone proprietà e implicazioni pratiche;
- sviluppare attenzione e cura, inclusa una certa sensibilità per l'eleganza, nella formalizzazione di strategie algoritmiche attraverso un linguaggio di programmazione;
- sviluppare competenze ai fini dell'organizzazione di un programma in unità funzionali.

Come già accennato, i principali prerequisiti per trarre profitto dalla partecipazione al campus, sono: interesse per la programmazione, curiosità per la ricerca di soluzioni algoritmiche (di problemi) eleganti e "ben congegnate", conoscenza delle basi di un qualsiasi linguaggio di programmazione.

La partecipazione degli studenti sarà inoltre riconosciuta nell'ambito dei Percorsi per le Competenze Trasversali e per l'Orientamento (PCTO – ex Alternanza Scuola-Lavoro).

### Informazioni pratiche

L'iniziativa potrà accogliere fino a 50 partecipanti. Nel caso di un numero eccedente di candidature, gli organizzatori effettueranno una selezione sulla base dell'applicazione bilanciata di due parametri: la distribuzione fra istituti diversi e l'ordine cronologico in cui le candidature saranno pervenute via e-mail all'indirizzo [cort@uniud.it](mailto:cort@uniud.it). L'accoglimento delle candidature verrà comunicato all'insegnante referente entro il 4/06/2021.

Per comunicare l'interesse alla partecipazione è sufficiente inviare all'indirizzo [cort@uniud.it](mailto:cort@uniud.it) una e-mail con allegato il [modulo di adesione](#), disponibile attraverso le pagine online del PLS, contenente le seguenti informazioni:

1. Istituto proponente;
2. Nominativi e recapiti e-mail di allieve e allievi che si intende candidare;
3. Nominativo di un/a insegnante referente per comunicazioni organizzative.

Ciascun/a partecipante al Campus dovrà preliminarmente installare su una propria postazione di lavoro

- il client Microsoft Teams, scaricabile gratuitamente attraverso l'URL:  
[www.microsoft.com/it-it/microsoft-teams/download-app](http://www.microsoft.com/it-it/microsoft-teams/download-app)
- l'ambiente didattico di programmazione in Java "BlueJ" che sarà utilizzato durante il Campus:  
[www.bluej.org](http://www.bluej.org)

I partecipanti apprenderanno l'uso del linguaggio di programmazione Java e dello strumento BlueJ nel corso delle attività introduttive del Campus.

### Contatti per informazioni o chiarimenti

Claudio Mirolo ( [claudio.mirolo@uniud.it](mailto:claudio.mirolo@uniud.it) ) e Emanuele Scapin ( [emanuele.scapin@uniud.it](mailto:emanuele.scapin@uniud.it) )

presso: Dipartimento di Scienze Matematiche, Informatiche e Fisiche dell'Università di Udine

Aggiornamenti del programma e informazioni organizzative saranno comunicati attraverso l'URL:

[nid.dimi.uniud.it/projects/pls\\_18\\_21.html](http://nid.dimi.uniud.it/projects/pls_18_21.html)

**Programma di massima delle attività** (ci potranno essere variazioni in alcuni contenuti e/o relative all'ordine delle attività)

	ore 9:30–12:30	ore 14:30–17:30
Lunedì 14/06/2021	<p>Presentazione degli obiettivi e delle attività del Campus e test preliminare di comprensione dei costrutti della programmazione strutturata.</p> <p>Introduzione ai costrutti imperativi di Java e all'ambiente di sviluppo BlueJ.</p> <p>Attività pratica di programmazione: sviluppo di semplici programmi in Java.</p>	<p><b>Talk – Introduzione alla codifica di Huffman: caratterizzazione algoritmica e proprietà generali.</b></p> <p>Attività pratica di programmazione: esplorazione delle caratteristiche del toolkit preparato per facilitare l'accesso ai file.</p>
Martedì 15/06/2021	<p>Attività pratica di programmazione: rilievo delle frequenze dei caratteri che occorrono in un documento applicando il toolkit.</p> <p>Introduzione ai costrutti base per la definizione di classi e oggetti in Java.</p> <p>Attività pratica di programmazione: rappresentazione dell'albero di Huffman in Java.</p>	<p><b>Talk – Intervento sulle tecniche di compressione lossy: codifica di immagini in formato JPEG.</b></p> <p>Introduzione all'uso di classi delle API Java e alla struttura dati coda con priorità.</p> <p>Attività pratica di programmazione: applicazione di una coda con priorità per costruire l'albero di Huffman.</p>
Mercoledì 16/06/2021	<p>Introduzione alla visita ricorsiva di un albero.</p> <p>Attività pratica di programmazione: generazione ricorsiva della tabella dei codici di Huffman.</p> <p>Attività pratica di programmazione: generazione di un file compresso utilizzando i codici di Huffman.</p>	<p><b>Talk – Ottimalità dell'albero di Huffman e "invarianti" relativi al processo per costruirlo.</b></p> <p>Attività pratica di programmazione: decodifica dei codici letti dal file compresso.</p> <p>Attività pratica di programmazione: ripristino del documento originale a partire dal file compresso e dall'albero di Huffman.</p>
Giovedì 17/06/2021	<p>Attività pratica di programmazione: codifica sequenziale e decodifica (ricorsive) dell'albero di Huffman.</p> <p>Attività pratica di programmazione: integrazione del file compresso con informazioni per il ripristino del documento originale.</p>	<p><b>Talk – Trasformazione della ricorsione in iterazione utilizzando uno stack (pila): "invarianti" del processo in relazione al contenuto dello stack.</b></p> <p>Attività pratica di programmazione: messa a punto degli strumenti per la compressione e il ripristino di un documento.</p>
Venerdì 18/06/2021	<p>Introduzione all'uso della struttura dati stack.</p> <p>Attività pratica di programmazione: trasformazione degli schemi ricorsivi del programma di compressione in schemi iterativi basati sullo stack.</p> <p>Microprogetti per la verifica dell'apprendimento.</p>	<p><b>Talk – Intervento su altre tecniche di compressione lossless.</b></p> <p>Microprogetti per la verifica dell'apprendimento (continuazione).</p> <p>Conclusioni del Campus.</p>

**Note biografiche sui relatori**

**Claudio Mirolo** è ricercatore di Informatica presso il Dipartimento di Scienze Matematiche, Informatiche e Fisiche dell'Università di Udine, dove attualmente insegna programmazione, geometria computazionale e didattica dell'informatica. Collabora con il Laboratorio nazionale CINI "Informatica e Scuola". I suoi principali interessi di ricerca riguardano l'apprendimento della programmazione e il ruolo del "pensiero computazionale" a diversi livelli di istruzione.

**Alberto Policriti** è professore di Informatica presso il Dipartimento di Scienze Matematiche, Informatiche e Fisiche dell'Università di Udine. I suoi principali interessi di ricerca comprendono aree della logica applicata all'informatica, dell'algoritmica, della verifica di modelli, della biologia computazionale e della bioinformatica. È titolare di corsi di Algoritmi e Strutture Dati, Algoritmi Avanzati e Bioinformatica. Ha recentemente collaborato a un percorso formativo sul pensiero computazionale, organizzato dall'Università di Pisa e rivolto agli insegnanti.

**Emanuele Scapin** è insegnante di Informatica presso l'ITT "Giacomo Chilesotti" di Thiene (VI), con una lunga esperienza didattica, e attualmente dottorando presso il Dipartimento di Scienze Matematiche, Informatiche e Fisiche dell'Università di Udine con un progetto di ricerca in didattica dell'informatica. Collabora all'insegnamento nell'ambito del Laboratorio del corso di Programmazione, rivolto agli studenti del primo anno dei due corsi di laurea in discipline informatiche con sede a Udine.

**Elio Toppano** è professore di Informatica presso il Dipartimento di Scienze Matematiche, Informatiche e Fisiche dell'Università di Udine, nell'ambito del quale attualmente è titolare dei corsi di Tecnologie Web e Progettazione di Sistemi Multimediali. Ha svolto le principali attività di ricerca nei settori dell'Intelligenza Artificiale, della Progettazione Ipermediale e dell'Ingegneria Ontologica.