Indice delle domande degli esami orali: Ingegneria Informatica LM

Questo file contiene le testimonianze degli esami orali di vari studenti del corso di laurea in **Ingegneria Informatica Laurea Magistrale** all' **Unical** (*Università della Calabria*) e fa parte del progetto Indice Argomenti Orali gestito dall'organizzazione **UnicalLoveTelegram**

Leggi il nostro README per conoscere tutti i dettagli del progetto, sapere come partecipare e come sfogliare tutto il nostro materiale!

- Sistemi Distribuiti e Cloud Computing (6 CFU e 9 CFU)
  - Talia Domenico
  - Loris Belcastro
- Basi di Dati evolute
  - Molinaro Cristian
- Calcolo Numerico
  - Yaroslav Sergeyev
  - Marat Mukhametzhanov
- Algoritmi di Crittografia
  - Cristian Molinaro

Data Mining

Sergio Greco

2021 2022

- Alessio
  - clustering gerarchico
  - entropia
  - reti neurali

Sistemi Informativi Ambientali]

Davide Luciano De Luca

2021 2022

- giovixo97
  - cosa è un DEM?
  - differenza file shape vettoriale e file raster
  - tecniche di geoprocessing
  - tutti i tipi di interpolatori (esatto,non esatto,locale,globale...)
  - cosa vuol dire la media o la varianza in un certo punto?
  - cosa rappresenta Z0? Ponendo Zi come i punti che hanno misura esatta con i>0
  - cosa è una misura?
    - lo strumento misura sicuramente bene
  - cosa è un GCP?
  - come è fatto un file di tipo geografico?
    - numero delle righe,colonne,risoluzione,xcornern,ycornern...

Ethical Hacking

Francesco Lupia

2020 2021

- Anonimi
  - Reverse Shell e Bind Shell

    - sql injection con script php (cosa è e cosa fa)
    - challenge web con loose comparison
    - differenze attacchi x32 bit e x64 bit
        - rop chain e bruteforce sul indirizzo di ritorno
    - Metasploit cosa è
    - tool simili a metasploit per windows
    - challenge web che presentava degli endpoint e bisognava loggarsi come admin
    - challenge web con form di login e registrazione
    - format string
    - privilege escalation windows: cosa faresti?

2021 2022

- Anonimi
    - Spiegazione csrf
    - Differenze tra csrf e xss
    - Cos'è kerberos
    - challenge SSRF presente sul sito di burp suite https://portswigger.net/burp (in teoria vi registrate, andate in accademy e poi nei vulnerabilty lab e cercate ssrf)
    - pass the hash: descrizione
    - challenge presente su natas numero 8 https://overthewire.org/wargames/natas/
    - Hash md5: come si riconosce?
    - siamo con una Macchina Windows e si devono rispondere alle seguenti domande poste dal prof:
        - psexec
        - pass the ticket
        - comandi vari del prompt o powershell
        - rogue potato (e in generale da tenere sott'occhio qualsiasi cosa che sia potato, quindi juicy potato, hot potato...)
        - si hanno degli output di eseguibili di Windows (permessi di un eseguibile e le info di un eseguibile) e fra i permessi di questo eseguibile c'era gitconfig e si poteva cambiare la configurazione per cambiare il /bin/path con una reverse shell
        - query su un registro per vedere se era attivo il permesso su un utente (alwaysinstalledprivileged) e si poteva sfruttare per installare qualsiasi eseguibile come utente privilegiato
        - pass the hash
        - bind e reverse shell
        - nishang

Angelo Furfaro

2021 2022

- Anonimi
    - Kerberos: cosa è ed attacchi
    - Docker:cosa è, configurazioni e comandi, attacchi (soprattutto privilege escalation)
    - Parte di privilege escalation disponibile su tryhackme (privesc)
    - Metasploit: come usare i servizi e gli exploit
    - nc: cosa è e come funziona
    - XXE: cosa è, scenari d'uso, esempi
    - XXS: cosa è, quali tipi ci sono, esempi
    - Laboratorio di attacco
    - Sudo con opzione -l
    - Utente con Alcuni privilegi di root
    - LDAP nel particolare
    - come creare una sottorete con virtualbox e come collegare due macchine alla sottorete

- come creare un laboratorio con virtualbox

Metodi Informatici per l'analisi dei Processi

Antonella Guzzo

2020/2021

- Anonimi
  - C-Net vs Heuristic net
  - Petri net Vs heuristic net
  - come viene fatta la classificazione delle attività iniziali e finali su ProM
  - workflow net (definizione)
  - cos'è la threshold
  - betweeness Nella resource analysis
  - differenze fra pattern merge e discriminator (bpmn)
  - perché scegliere un modello (o un plugin) rispetto ad un altro
  - boundness
  - quando il marking è dead?
  - esercizi su boundness e deadlock
  - alpha miner (con i vari punti specifici)
  - qualità del modello
  - in cosa consiste la classificazione di un dato
  - perché è costoso l'alpha miner?
  - domande sul progetto in generale e nello specifico
  - liveness
  - come ottenere un buon modello?
  - conformance e tipologie

Metodi e Strumenti per la Sicurezza Informatica

Michele Ianni

2020 2021

- Giovanni Giordano
  - http://basicrce.challs.cyberchallenge.it/ risolvi la challenge edit: è andato down, la challenge consisteva in un form html che faceva una post all'indirizzo /ping dello stesso sito e ritornava semplicemente il codice di ritorno della shell linux collegata e il comando eseguito, altrimenti dava errore. Non c'era nient'altro, bisognava trovare la flag.txt da qualche parte nel sito.
  - GOT e PLT
- Anonimi
  - Canary
  - gdb
  - sito che ritorna un immagine, come capisci le tabelle?
  - nmap port scanning
    - fin scan
    - udp scan
    - syn scan
    - null scan
    - xmas scan
  - arp poisoning
  - reflected, DOM Based e stored XSS

- ASLR
- CSRF
  - chi genera il token
- ROP
  - come mai i tool automatizzati trovano tanti gadget mentre una scansione manuale ne trova pochi?
  - i gadget sono una serie di istruzioni. Perché ropper va a guardare l'esadecimale, parte da una ret e va all'indietro se una sotto sequenza è un'istruzione valida viene restituito il gadget. Ad esempio in esadecimale a3 aa bb cc 90 c3 è mov eax, 0x90aabbcc; ret, ma la sottosequenza 90 c3 è nop; ret. Sono entrambi gadget.
- buffer overflow
  - mitigazioni
  - generarlo senza utilizzare le funzioni vulnerabili
- code reuse
- Mitigazioni SQL injection

2021 2022

- Anonimi
  - format string
  - xss
    - le differenze tra i vari tipi di xss
  - ARP poisoning
  - port scanning
    - FIN SCAN
    - XMAS SCAN
    - SYN SCAN
  - ret2libc
    - perché è meno conveniente rispetto alla code reuse?
  - Plt e got
  - Xss
  - Canary
    - perché si usa il carattere 0
  - Ropper
  - Blind sql injection

Business Intelligence

Filippo Furfaro

2020 2021

- Anonimi
  - gestione delle dimensioni degeneri
  - gerarchie dinamiche
  - a cosa serve attributo master nello scenario di verità storica
  - a cosa servono le chiavi surrogate
  - perchè non si usano i btree
  - star index
  - join index
  - quando conviene fare snow flake
  - gerarchie incomplete e soluzioni

    - indici bitmap a confronto con btree
    - molap e rolap
    - Tutti i pro e tutti i contro dell'usare Chiavi surrogate
    - Star index
        - quando non è efficiente usare lo star index
    - aggregatori olistici
    - indici di bit-sliced
    - gerarchie ricorsive (pro e contro delle 2 soluzioni)

Strategie e Politiche Aziendali

Patrizia Pastore

2020 2021

- Anonimi
    - cosa faresti da imprenditore della tua azienda (cyber security), ovvero quali strategie sceglieresti tra quelle viste nel corso
    - classificazione outsourcing
    - scelta di un settore in cui competere e forze di porter
    - esempi a lezione
    - la valutazione comprende i punteggi dati al test online di fine corso (crocette) e i lavori in ppt di gruppo
    - Stakeholder amichevoli
    - Outsourcing
    - Finalitá dell azienda

Modelli e Tecniche per i Big Data

Paolo Trunfio

2020 2021

- Anonimi
    - parametri mpi speedrun tempo esecuzione parallelo e sequenziale
    - lambda expression
    - benefici java stream
    - differenze spark hadoop
    - RDD
    - hama
    - costo del calcolo bsp
    - zookeper
    - trajectory discovery
    - java stream lazy
    - legge amdhal
    - wordcount
    - mapper e reducer
    - spark e hadoop convenienza
    - bsp in generale
    - send receive non blocanti e bloccanti
    - spark lazy execution
    - wordcount reverse (chiave lunghezza parole)
    - logica di hive
    - legge di amdhal

    - comunicazione in MPI sincrona e asincrona e meccanismi
    - caratteristiche di un programma in parallelo
    - combiner in mapreduce
    - numero di reducer e mapper
    - watermark
    - wordlenghtcount

2020 2021

- Anonimi
    - codice word count
    - che tipologia di programmi esegue storm
    - possono esserci piu spout?
    - quali metodi deve implementare spout e quali bolt
    - combiner di map reduce
    - codice word count reverse
    - Superlinear speedup:
    - architettura hdfs e file di configurazione delle risorse

Architetture e programmazione dei sistemi di elaborazione

Fabrizio Angiulli

2016 2017

- Roberto
    - cache completamente associativa
    - open MP
    - schema monociclo e segnali di controllo +1
    - cache a k vie
    - multithreading
    - grana fine
    - grana grossa
    - vantaggi multithreading simultaneo (ogni thread a i suoi registri e PC)
    - differenza multithreading sw e multithread hw
    - dimensionamento clock multicolore
    - conflitti sul controllo
    - statistica a 2 bit automa
    - nano programmazione
    - emissione fuori ordine
    - tabella segnali alpha monociclo
    - conflitti sui dati pipeline
    - conflitti superscalari
    - ottimizzazione unità di controllo (control store)
    - completamente fuori ordine e ritiro in ordine
    - CPU vs GPU
    - una numa
    - macchina multiciclo
    - macchina monociclo
    - dimensionamento del clock della multi ciclo
    - ottimizzazione della parte di controllo microprogrammata
    - legge di moore e barriera dell'energia
    - speculazione nell'hardware
    - speculazione hw (epr)
    - buffer di ordinamento macchina super scalare
    - completamento fuori ordine

    - emissione fuori ordine
    - numero di posizioni
    - ottimizzazione del controllo microprogrammato
    - predizione dei salti schema
    - politiche sostituzione della cache
    - disegno
    - speculazione hardware macchina super scalare
    - differenza uma e numa
    - macchina hasswell
    - differenze cics e risc
    - principi di progettazione risc
    - riduzione parallela
    - rsr

2019 2020

- Anonimi
    - Legge di Moore e barriera energia
    - Macchina multiciclo
    - ottimizzazione unità di controllo (control store programmato)
    - Nano programmazione
    - dimensionamento del clock nella multi ciclo microprogrammata
    - differenze macchine cisc e risc
    - principi di progettazione macchina risc
    - schema monociclo e tabella segnali alpha
    - conflitti sui dati pipeline
    - emissione fuori ordine
    - Rsr
    - completamente ofuori ordine
    - ritiro in ordine
    - confliti sul controllo
    - predizione dei salti a schema - branch prediction unità
    - statistica a due bit con automa
    - conflitti sulle super scalari
    - buffer di ordinamento macchina super scalare
    - speculazione hardware (epr)
    - completamento fuori ordine macchina super scalare
    - Macchina di Haswell
    - cache completamente associativa
    - cache a k vie
    - politiche di sostituzione nella cache disegno
    - differenza uma e numa
    - multithreading hw : grana fine e grana grossa
    - vantaggi multithreading simultaneo
    - differenza multi threading hw e sw
    - cpu vs gpu
    - riduzione parallela
    - open mp
- Giovanni giordano
    - cache a k vie
    - cache a mappatura diretta
    - tipi di threading
    - conflitti pipeline

2020 2021

- Erma_TV
    - conflitti sulla pipeline quali sono e come si risolvono
    - CISC RISC
    - principi dei modelli di calcolatori di oggi

        - UMA e NUMA con disegno della NUMA
        - speculazione hardware come avviene e dove avviene
        - attacco spectr
        - c'é speculaizone hardware nella pipeline? No, come vengono gestiti i salti?
- Anonimi
    - Cache
    - Politiche di sostituzione
    - Unità di controllo monociclo
    - Segnali beta mono e multi
    - Ottimizzazione controllo micro programmato
    - Circuito di selezione degli indirizzi
    - Disegno stack lru
    - E disegno circuito di selezione degli indirizzi
    - Ottimizzazione controllo microprogrammato
    - Macchine parallele
    - Nanoprogrammazione
    - circuito propagazione nella superscalare
        - circuito di bypass
    - NUMA e UMA
    - conflitti sul controllo
    - conflitti nella pipeline: inserimento circuito di uguaglianza
    - Confronto prestazionale fra tutte le macchine viste nel corso
    - Clock fine
    - Speculazione hw e cosa cambia rispetto alle predizioni della pipeline
    - Cache multilivello e come cambia il calcolo del tempo medio di accesso alla memoria

Crittografia e analisi reti sociali

Molinaro Cristian

2016 2017

- Tassone
    - Cifrario a flusso
    - OTP
    - PRG
    - Shannon
    - Cifrari a blocchi
    - Sicurezza semantica
    - PRP
    - ECP
    - CBC
    - CBC+nonce
    - CTR
    - CTR+nonce
    - MAC (funzionamento sicurezza e challange)
    - NMac
    - PMAC
    - HMAC
    - ECBC MAC
    - PAYLOAD

    - HASH (funzionamento sicurezza e challange)
    - PAradosso compleanno + attacco hash (collissioni)
    - Merkle damgard
    - Autenticazione cifrata (funzionamento sicurezza e challange)
    - tre tipologie costruzione autenticazione cifrata (e than m, e and m, m then e) più differenze e sicurezza
    - differenza chiave simmetrica e asimmetrica
    - principi chiave asimmetrica
    - RSA
    - Complessità attacco RSA per scoprire chaive segreta
    - complessita attacco RSA per un messaggio cifrato (differenza con sopra)
    - Merkle puzzle
    - autorità di certificazione e firma digitale (molto in generale più schema)
- Riccardo
    - generazione rsa per calcolo chiavi
        - come si cifra
        - come si decifra
    - rabin come si generano le chiavi
        - collegarsi alla fattorizzazione
        - output di 4 messaggi
        - cattiva proprietà del sistema
    - ElGamal su cosa è basato
        - come si calcolano le chiavi
    - tutti i possibili attacchi di chiave che si muovono contro RSA
        - brute force
        - euclide
        - vari problemi
    - puzzle di merkle
    - introduzione key managment e scneari utilizzo rsa

Linguaggi Formali

Domenico Saccà

2016 2017

- PsykeDady
    - Compilazione della tipizzazione dinamica dei linguaggi
    - tipizzazione dinamica che tipo di linguaggio è (risp: 2)
    - cos'è un automa a pila
- Marco Domenicano
    - Tautologia
    - conraddizione
    - memorizzazione di un json in calculista
    - esercizio del minimo locale in calculist e prolog
- Anonimi
    - come vengono memorizzati i json in memoria nella calculist

2019 2020

- Alfredo
    - json
    - linguaggi di primo, secondo e terzo tipo
        - java di che tipo è
        - html di che tipo è
        - xml di che tipo è

- Giovanni Giordano
  - calculist esercizio `Unione(L1,L2,L3)`
    - costruire L3 **unendo L1 e L2**
- Angelo
  - Scrivere automa a stati finiti deterministico che riconosce il linguaggio `(a+b+)+b*c`
    - fare esempio di una stringa che non appartiene al linguaggio
    - fare esempio di stringa che appartiene al linguaggio
- Anonimi
  - Calculist esercizio `Intersezione(L1,L2,L3)`
    - costruire L3 come **intersezione di L1 e L2**
  - cos'è un modello logico
  - quando un modello è minimo
  - Calculist lista ordinata L
  - Calculist High Order Function espressione con lambda function
  - complessità del problema di stabilire se un programma logico ammette un unico modello (sol. *PSPACE*)
  - Verificare se due Liste L1 e L2 hanno gli stessi elementi

2020 2021

- Anonimi
  - high order function
  - solito esempio con u(X),p(X),r(X),rc(X)
  - universo di Herbrand, Base di Herbrand, modelli minimali
  - verificare che 2 liste abbiano gli stessi elementi con lo stesso numero di occorrenze
  - espressioni regolari
  - unificatore generale
  - Palindroma in Calculist

Rullo

2016 2017

- Marco Domenicano
  - scrivere un programma in prolog che riceve una lista L, T, T1 e restituisce una lista di copia in output L1 così composta: se elemento di L corrisponde a T inserisci T1 altrimenti L

2019 2020

- Alfredo
  - 2 esercizi prolog
- Giovanni Giordano
  - esercizio prolog su traccia `P(L1,L2,L3,L4)`, soddisfare:
    1. `L3 come L1 intersecato L2`
    2. `L4 come L1 - L2`
  - esercizio prolog su traccia su traccia `P(T,T1,L,L1)`, soddisfare
    - `se L[i]≠T verificare L[i]==L1[i] altrimenti L1[i]==T1`
- Angelo
  - scrivere un metodo int(L1,L2,L3) che restituisce vero se:
    1. L1 sotto insieme improprio di L3
    2. L2 sotto insieme improprio di L3
    3. L3 non contiene duplicati
    4. L1,L2,L3 sono ordinati in modo crescente
- Anonimi
  - scrivere un programma prolog che: `dati due termini T e T1 e una lista L`

        - produce una lista L1 identica a L in cui sono state sostituite tutte le istanze di T con T1, ossia la relazione `subst(T,T1,L,L1)` dove L1 è la lista ottenuta da L sostituendo tutte le istanze del termine T con T1 lasciando gli altri elementi invariati
        - `p(L1,L2)` che restituisce true se L1 ed L2 contengono gli stessi elementi
    - lanciare la computazione in calculist
        - descrivere stato memoria
        - dare risultato
    - Teorema di Rice (accenno)
    - quanti sono i modelli di un programma positivo
    - cos'è l'unificazione di due termini?
    - data:
        - `g(x/2)/1: lambda z: x(y,z+y);`
        - eseguire: `g(molt,3)(4);` risultato?
    - Quanti modelli minimali ci sono in questo programma logico?

```
u(1).
u(2).
u(3).
p(1).
p(2).
r(X):
u(X), not(p(X)).
rc(X):- u(X), not(r(X)).
g(x/2,y)/1: lambda z: x(y,z+y);
pp(x,y): x+2*y;
^g(pp,3)(4);
```

- - risultato=17
    - quanti sono i modelli minimali (stesso modello)?
        - `u(1).`
        - `u(2).`
        - `p(1).`
        - `r(X):- u(X), not(p(X)).`
        - `rc(X):- u(X), not(r(X)).`
    - cos'è un universo
        - tutti i termini ground, nel caso di prima i primi due
    - funziona calculist che dato `x` calcola `fibonacci(x)`
    - dato:

```
u(1).
u(2).
p(1).
r(X):- u(X), not(p(X)).
rc(X):- u(X), not(r(X)).
```

- - quanti sono i modelli minimali
        - **Legenda**: u sono gli umani, p sono i poveri, r è una persona ricca, rc è il reddito di cittadinanza (i significati hanno poca rilevanza).
        - **Risposta**: quando si ha la negazione di solito si hanno piu modelli minimali
        - **modello migliore**: `rc(X)=true` solo in un caso (reddito di cittadinanza solo ad un elemento)
    - scrivere un metodo che riceve in ingresso 4 liste `q(L1, L2, L3, L4)` che restituisce `true` se **L3** è l'itersezione di **L1+L2** ed **L4=L1-L2** (sottrazione insieimistica), le liste vanno intese come insiemi.
    - scrivere un metodo `q(A,B,L1,L2)` che restituisce true `L1=L2` con i caratteri **A sostituiti con B in L2**
    - scrivere un `q(X,L,Y)` che restituisce vero se **Y** è l'elemento successivo a **X** nella **L**
    - scrivere un `q(X,L,Y)` che restituisce vero solo se **Y** è nella posizione **X** di **L**

2020 2021

- Anonimi
  - riceve 2 liste: true se le due liste contengono gli stessi elementi, anche con numero di occorrenze diverso
  - ricerca binaria in prolog
  - Scrivere un programma Prolog che, dati due termini T e T1 e una lista L, produce una lista L1 identica a L in cui sono state sostituite tutte le istanze di T con T1, ossia la relazione: subst(T,T1,L,L1), dove L1 è la lista ottenuta da L sostituendo tutte le istanze del termine T con il termine T1 e lasciando invariati gli altri elementi p(1,2,[1,1,2,2], [2,2,2,2])
  - Si scriva un programma Prolog che, prendendo in ingresso due liste L1 e L2, restituisca in uscita due liste L3 e L4 tali che L3 contenga gli elementi di L1 che appartengono anche a L2, mentre L4 contenga gli elementi di L1 che non appartengono a L2. Si supponga disponibile il predicato member p([a,r,t],[t,s,m,n,a],L3,L4) p([a,r,t],[t,s,m,n,a], [a,t] ,[r])
  - Scrivere un programma PROLOG per la seguente relazione: d(X,Y) se e solo se Y è la lista che si ottiene dalla lista X rimuovendo gli elementi di posizione pari
  - Define a predicate add_up_list(L,K) which, given a list of integers L, returns a list of integers in which each element is the sum of all the elements in L up to the same position. add_up_list([1,2,3,4], [1,3,6,10])
  - Scrivere un programma Prolog che, dati due termini T e T1 e una lista L, produce una lista L1 identica a L in cui sono state sostituite tutte le istanze di T con T1, ossia la relazione: subst(T,T1,L,L1), dove L1 è la lista ottenuta da L sostituendo tutte le istanze del termine T con il termine T1 e lasciando invariati gli altri elementi
  - Definire il predicato Prolog fib(N,F) che sia vero se F rappresenta l'N-esimo numero della sequenza di fibonacci. Ricordiamo che la sequenza di Fibonacci è definita dalle seguenti: f(0) = 1, f(1) = 1 ,f(N) = f(N - 1) + f(N - 2)
  - Si scriva un programma Prolog che, prendendo in ingresso due liste L1 e L2, restituisca in uscita due liste L3 e L4 tali che L3 contenga gli elementi di L1 che appartengono anche a L2, mentre L4 contenga gli elementi di L1 che non appartengono a L2. r([1,2,3],[3,4,5,6,1],L3,L4)
  - Define a predicate reverse(L,K) which holds if and only if the list K is the reverse of the list L
  - Define a predicate occurs(L,N,X) which holds iff X is the element occurring in position N of the list L
  - Define a predicate add_up_list(L,K) which, given a list of integers L, returns a list of integers in which each element is the sum of all the elements in L up to the same position. Example: ?- add_up_list([1,2,3,4],K). K = [1,3,6,10]
  - Define a predicateoccurs(L,N,X)which holds iffXis the element occurring in positionNof the listL
  - palindroma
  - Scrivere un programma Prolog che, dati due termini T e T1 e una lista L, produce una lista L1 identica a L in cui sono state sostituite tutte le istanze di T con T1, ossia la relazione: subst(T,T1,L,L1), dove L1 è la lista ottenuta da L sostituendo tutte le istanze del termine T con il termine T1 e lasciando invariati gli altri elementi
  - Si scriva un programma Prolog che, prendendo in ingresso due liste L1 e L2, restituisca in uscita due liste L3 e L4 tali che L3 contenga gli elementi di L1 che appartengono anche a L2, mentre L4 contenga gli elementi di L1 che non appartengono a L2. Si supponga disponibile il predicato member.
  - Define a predicate occurrences(X,L,N) which holds iff the element X occurs N times in the list L
  - Definire il predicato Prolog fib(N,F) che sia vero se F rappresenta l'N-esimo numero della sequenza di fibonacci. Ricordiamo che la sequenza di Fibonacci è definita dalle seguenti: f(0) = 1, f(1) = 1, f(N) = f(N - 1) + f(N - 2)
  - Scrivere un programma PROLOG per la seguente relazione: d(X,Y) se e solo se Y è la lista che si ottiene dalla lista X rimuovendo gli elementi di posizione pari.
  - Define a predicate add_up_list(L,K) which, given a list of integers L, returns a list of integers in which each element is the sum of all the elements in L up to the same position
  - Define a predicate merge(L,K,M) which, given two ordered lists of integers L and K, returns an ordered list M containing all the elements of L and K
  - dd(f/2,x)/1: lambda y: f(y)+2*x: s2(x): 2* x; ^dd(s,3)(4); funzione lambda proposta

2021 2022

- Anonimi
  - Scrivere un programma Prolog che, dati due termini T e T1 e una lista L, produce una lista L1 identica a L in cui sono state sostituite tutte le istanze di T con T1, ossia la relazione: subst(T,T1,L,L1), dove L1 è la lista ottenuta da L sostituendo tutte le istanze del termine T con il termine T1 e lasciando invariati gli altri elementi
  - Si scriva un programma Prolog che, prendendo in ingresso due liste L1 e L2, restituisca in uscita due liste L3 e L4 tali che L3 contenga gli elementi di L1 che appartengono anche a L2, mentre L4 contenga gli elementi di L1 che non appartengono a L2
  - stessiElem(L1,L2), which holds if L1 and L2 have same elements
  - Define a predicate occurrences(X,L,N) which holds iff the element X occurs N times in the list L

- Scrivere un programma Prolog che, dati due termini T e T1 e una lista L, produce una lista L1 identica a L in cui sono state sostituite tutte le istanze di T con T1, ossia la relazione: subst(T,T1,L,L1), dove L1 è la lista ottenuta da L sostituendo tutte le istanze del termine T con il termine T1 e lasciando invariati gli altri elementi
- Si scriva un programma Prolog che, prendendo in ingresso due liste L1 e L2, restituisca in uscita due liste L3 e L4 tali che L3 contenga gli elementi di L1 che appartengono anche a L2, mentre L4 contenga gli elementi di L1 che non appartengono a L2.
- Define a predicate occurs(L,N,X) which holds iff X is the element occurring in position N of the list L.
- Define a predicate add_up_list(L,K) which, given a list of integers L, returns a list of integers in which each element is the sum of all the elements in L up to the same position

Informatica teorica

Scarcello Francesco

2016 2017

- PsykeDady
  - Teorema di Cook
  - Definizione di NP complete
- Riccardo
  - Partendo dal fatto che un problema è np-hard se qualsiasi problema np si riduce ad esso in tempo polinomiale
    - domanda: come cambia la classe np-complete se cambiamo la definizione di hardness considerando trasformazioni esponenziali invece che polinomiali?
    - risposta: Poiché np-complete è l'intersezione di np-hard ed np, i problemi di tale classe rappresentano il sottoinsieme dei problemi più difficili tra quelli appartenenti ad np (risolvibili in p-time da una NTM). Se si cambia la definizione di hardness considerando trasformazioni esponenziali però si estende la classe a problemi exp-time, in quanto si altera il rapporto di complessità durante la riduzione che supporta la hardness: intuitivamente, una trasformazione esponenziale trasferirebbe parte della complessità nella riduzione, permettendo poi di risolvere il problema risultante in tempo polinomiale, dunque tali problemi ricadrebbero in questa versione modificata di np-complete.
- Anonimi
  - Teorema di Cook
  - Definizioni di problema Np, Np-hard, Np-complete
  - Dimostrazione di appartenenza di Hamiltonian Cycle a Np-Complete
  - Dimostrazione di non appartenenza di Ld a RE
  - Dimostrazione di appartenenza di Lu a RE
  - Definizione di riduzione
  - Teorema di Rice

2017 2018

- Marco
  - Linguaggio Empty
  - dimostrazione NP complete
  - dimostrazione indipendent Set

(continuare da 2016 2017 linguaggi formali sacca psykeS)

2018 2019

- Matteo Grollino
  - Teorema Rice
  - Teorema Cook
  - Knapsack Intero e Frazionario
  - subset sum
  - approssimabilità knapsack

        - Algoritmo pseudo-polinomiale
        - FPTAS
    - Definizione NP
    - Definizione NP Hard
    - Definizione NP Complete
    - Dimostrazioen indecidibilità Lu e non appartenenza a RE di Ld
    - Importanza riduzione polinomiale tra problemi decisionali
    - Perché NP è incluso in PSpace con dimostrazione
    - complessità parametrizzata con definizione di XP e FP
    - Algoritmo FPT del vertex Cover
- Gianpaolo
    - Teorema 4.14.1 : un problema NP ha come definizione NP = {L|E R polinomialmente decidibile e bilanciata che caratterizza L } con Pl1 R=L (dimostrazione)

2019 2020

- Angelo
    - definizione di problema np-completo
    - cos' é una trasformazione polinomiale?
    - dimostrazione del teorema di Rice
    - fixed parameter trattability
    - cos' é uno schema di approssimazione polinomiale ?
    - dimostrare che nap-sack é np-hard
    - perché usiamo trasformazioni polinomiali e non esponenziali?
    - dimostrare che ld é ricorsivamente enumerabile
    - definizione di np-hard
    - dimostrare che Hamiltonian cycle é np-hard
- Giovanni Giordano
    - Dimostrazione linguaggio NTM==DTM
    - caratterizzazione NP dimostrato
    - Indipendent Set dimostrato
- Anonimi
    - cook
    - NP dentro PSpace (dimostrazione)
        - **Risposta**: Perchè la definizione di NP dice che NP appartiene a Ptime, poichè Ptime è un sottoinsieme di Pspace allora anche NP è un sottoinsieme di Pspace
    - teorema di Rice
    - np completo (definizione) e vantaggi nellúso
    - Teorema di Cook
    - Definizione di problema NP-complete
    - Domanda: `come cambia la clas shortcut multicursorsse np complete se cambiamo la definizione di hardness considerando trasformazioni esponenziali`
        - **Risposta**: poiché np-complete é l'intersezione di np-hard ed np, i problemi di tale classe rappresentano il sottoinsieme dei problemi più difficili tra quelli appartenenti ad np (risolvibili in p-time da una NTM). Se si cambia la definizione di hardness considerando trasformazioni esponenziali però si estende la classe a problemi exp-time, in quanto si altera il rapporto di complessità durante la riduzione che supporta la hardness: intuitivamente una trasformazione esponenziale trasferirebbe parte della complessità nella riduzione, permettendo poi di risolvere il problema risultante in tempo polinomiale, dunque tali problemi ricadrebbero in questa versione modificata di np-complete.
    - Dimostrazione di appartenenza di Hamiltonian Cycle a np-complete
    - dimostrazione di non appartenenza di Ld a RE
    - Dimostrazione di appartenenza di Lu a RE
    - definizione di riduzione

    - Linguaggio Empty dimostrazione NP complete
    - mostrazione Indipendent SET
    - Knapsack intero e frazionario
    - subset sum
    - Approssimabilità knpasack (algoritmo pseudo polinomiale e FPTAS)
    - importanza della riduzione polinomiale tra problemi decisionali
    - complessità parametrizzata con definizione di xp e di ffpt
    - problema np ha come definizione NP = `{L| E R polinomialmente decidibile e bilanciata che caratterizza L}` con PI1 R=L (dimostrazione)
    - FPTAS con costi
    - FPT con VC e con knapsack
    - knapsack con programmazione dinamica

2020 2021

- Erma_TV
    - Dimostrazione NP incluso in PSPACE
    - Dimostrazione che Knapsack ammette un FPTAS
    - Che sono le classi di approssimabilitá
- Anonimi
    - Rice con dimostrazione
    - FPT
    - FPT con vertex cover (con le due soluzioni)
    - Dimostrare che Subset Sum è NP-Hard
    - Rice con dimostrazione
    - NL con dimostrazione che è NP-Hard
    - vertex cover
    - indipendet set
    - hamiltonian cycle
    - NTM = DTM
    - def di NP-complete (NP-HARD, NP)
    - L appartiene ad NP se e solo se esiste una relazione caratteristica RL di L (parte <=) e (parte =>)
    - Bisaccia FPTAS

Ottimizzazione

Maria Flavia Monaco

2016 2017

- PsykeDady
    - Argomento a piacere : Rilassato LaGrangiano
    - Definizione di problema Rilassato
    - Duale LaGrangiano (perché farlo? obiettivi)
    - Vehicle Routing Problem formulazione
- Anonimi
    - che ho a disposizione se voglio risolvere un problema piccolo con un algoritmo esatto ? (B&Bound)
    - Cosa si intende per "cut" e quindi un algoritmo di `branch and cut`
    - Gomory, tutto il procedimento
    - Perché posso usare la funzione obiettivo in gomory per indurre un taglio?
    - come si valuta un euristica? Lagrangiano
    - Definire duale di Lagrangiano

    - Commesso viaggiatore
        - come calcolo un lowerbound ?
        - perché non si usa Lagrangiano?
        - perché ha un numero esponenziale di cicli e molto probabilmente avrà sempre sottocicli
    - Problema del commesso viaggiatore non orientato
        - taglio con Branch and Cut
        - oracolo di Separazione
    - Formulazioni commesso viaggiatore sia orientato che non
    - Quando una formulazione è ottimale? (matrice TUM)
    - Per quale problema ho una formulazione ottimale anche se non è TUM? problema del matching
    - Set covering definizione
    - Commesso viaggiatore
        - perché è intrinsecamente combinatorio
        - complessità
    - come risolvo il set-covering (max saving)
    - chvatal
    - Vehicle routing
    - Algoritmo clarke wright (massimo risparmio)
    - Epsilon approssimativo
        - definizione
        - TSP
        - algoritmo dell'albero
    - Differenza Hamilton - eulero, con confronto tra i due
    - Teorema di minkowsky

2020 2021

- Anonimi
    - Set covering
    - Formulazione valida
    - ottima
    - Problema di localizzazione
    - Rilassamento lagrangiano
    - Se x è punto estremo => x appartiene ad S

Valutazione delle prestazioni

Pasquale Legato

2016 2017

- PsykeDady
    - problema del professore in ritardo (su excel)
    - produttore consumatore (excel)
    - modello di markov (slide)

Intelligenza Artificiale (6 CFU)

Palopoli Luigi

2017 2018

- PsykeDady
  - Estensione di Reiter
  - Anomalia di Sussman
  - breadth first (vantaggi rispetto a depth first)
  - strips
    - frame problem
    - quantification problem
    - representation problem
  - deep learning
    - definizione
    - reti neurali
    - struttura neurone
    - altri approcci
    - deep learning
    - features extracton
    - hill climbing + simulated annealing
    - pac learning
  - Anonime
    - IDA* perchè c'è min nella funzione
    - Frame assension
    - strips
      - risoluzioni
      - problemi del non essere linguaggio logico
    - estensione di reithers
    - come calcolarla
      - che succede se togliamo TH da IN(pigreco)
    - nucleolo

Intelligenza Artificiale e rappresentazione della conoscenza (12 CFU)

Palopoli Luigi

2019 2020

- Anonimi
  - Iterative Broadening (ordine di visita degli alberi)
  - Iterative Dipening
  - processi closed e successful
  - shapley value
  - wsat e gsat
  - estensioni di reiter
  - frame problem e perché strips non soffre del problema del frame
  - approssimazione lower bound-upperbound con calcolo greatest lower bound

2020 2021

- Anonimi
  - primo interrogato
    - hill climb simulated annealing
    - planning
    - nucleolo stable set
    - regole inferenza

        - entailment in logica di default perché è Pi P2-C?
        - gsat wsat con random walking
    - secondo interrogato
        - breadth first
        - Iterative broadening e come si fa con A*
        - Nucleolo di nuovo
        - Compilazione di conoscenza
        - datalog or not
    - terzo interrogato
        - metodi di ricerca blind e metodi di ricerca informata: differenze
        - iterative deepening con vantaggi
        - IDA*
        - semantica alla reiter default logic
        - semantica brave default logic
        - verifica coerenza teoria di default (NP Hard)
        - processo
        - nucleolo
    - quarto interrogato
        - iterative broadening
        - perché non usiamo A* per i giochi al posto di min max?
        - hill climb simulated annealing
        - modello stabile con negazione e disgiunzione
        - computer vision e algoritmo di waltz
        - planning
            - quale sequenza di azioni va considerata?
            - perché la delete list deve essere vuota?
        - stable set teoria giochi
        - N=1,2,3 v1=v2=v3=0 e la coalizione di taglia due hanno valore 2, la coalizione di tagla tre vale 5: c'è stable set?
    - quinto interrogato
        - metodi olistici di riconoscimento ambiente
        - pianificazione: Strips
            - Strips Assumption
            - A1:precondizione vuota, add list è P, delete list vuota,A2:precondizione vuota, add list not P, delete list vuota e stato iniziale vuoto. Risultato?
        - concetti soluzione che danno equità, Shapley Value
        - effetto orizzonte
        - singolar extension
        - nodo quieto e nodo tattico
        - A*
        - modello stabile per datalog not
            - intersezione tra modelli che provoca?
            - semantica modelli perfetti o modell stabili
    - sesto interrogato
        - test turing
        - regole di inferenza correttezza e completezza
            - Modus Ponens e completezza del modus ponens
                - esempio sound e non complete
            - quanto costa capire se f può essere generato da modus ponens con F?
            - versione arricchita del modus ponens Tp

        - di nuovo la cosa della add list di prima con riflessione su strips
        - waking sat
        - il numero dei GLB in una teoria CNF
        - bargening set
        - algoritmo della famiglia minmax a cui si applica alfa-beta con valori +0.001 e -0.001 in questo caso si taglia l'albero?
        - algoritmo waltz
    - settimo interrogato
        - numero GLB teoria di horn di dimensione n
        - come scende la complessità del caution reasoning?
        - pure theory
        - se una teoria ha un estensione non calcolabile attraverso i processi cosa succede?
        - A* con differenza best-first
            - la funzione euristica non esegue mai il backtracking?
        - Core
        - algoritmo waltz
    - ottavo interrogato
        - numero dei GLB? la congiunzione degli UB è 1 (unico LUB congiunto), anche la congiunzione dei GLB è pure 1 solo se la teoria è di horn (esponenziale se teoria default)
        - kernel
        - teoria di default che abbia un estensione che non possa essere calcolata dall'albero de processi?
        - IDA*
            - a cosa serve il min?
        - programma datalog stratificato
    - altri
        - Verie testimonianze 04/02/2021
        - Descrizione algoritmo Iterative deepening
        - Precisare come si può uscire dal ciclo quando non ci sono goal
            - **Risposta**: la soluzione proposta dal prof è quella di utilizzare una variabile booleana (non sappiamo nel dettaglio come), un'altra soluzione è quella di uscire quando il cutting level sia pari all'altezza dell'albero ma costa troppo in termini temporali
        - Complessità di verificare la coerenza di una teoria in logica di default (ossia se ammette un'estensione), dimostrare almeno intuitivamente perché tale problema è almeno NP-hard
            - **Risposta**: intuitivamente se la complessità dell'entailment è CONP-c in logica proposizionale, poiché la logica di default ha sia una teoria proposizionale W che un'insieme di default D è facile capire che sarà almeno difficile quanto l'entailment è quindi ha almeno una sorgente di esponenzialità
        - Strips genera stati inconsistenti?
            - **Risposta**:un esempio è {f, not(f)} in cui abbiamo uno stato con due fluenti con valore logico opposto, ma strips NON è un linguaggio logico, f e not f potrebbero essere chiamati pluto e paperino quindi no, non genera stati inconsistenti in quanto il concetto di incosistenza è associato a linguaggi logici)
        - Esempio di teoria di default in cui non ci sia alcuna estensione che sia calcolabile con la semantica operazionale
            - **Risposta**: basta usare una teoria incoerente, {TRUE:A/¬A } è l'esempio tipico
- Giovanni
    - GSAT
    - espressività vs complessità
    - hill climb con simulated annealing
    - modello perfetto

Sistemi Informativi

Cassavia

2017 2018

- Gianpaolo
  - Parte PENTAHO:
  - OLAP
  - modellazione concettuale data warehouse
  - realizzare in saiku roll up e roll down
  - document datastore
  - column family
- Luca
  - Creare in saiku l'operazione slice e selezione
  - modellazione logica dei data ware house
    - 4 fasi della modellazione
  - imputation mismatching
  - schema di HBase
    - disegnare
    - nome delle componenti
    - modi per interfacciarlo con il client
  - teorema CAP

2019 2020

- PsykeDady
  - presentazione progetto
  - eseguire su pentaho:
    - drill up
    - roll down
    - selection slice
  - fasi di progettazione Data Warehouse
  - Schemi di fatto a stella e snowflake
  - Proprietà sistemi nosql
  - utilizzo di hbase

ISSTRA Ingegneria del software per sistemi real-time ed agenti

Libero Nigro

2018 2019

- Anonimi
  - tempo di blocco FPS
  - conversione processo sporadico/periodico
  - Ping Pong in Jade
  - Grafo degli stati UPPAAL
  - Query In Uppaal
  - Scrivere un parcheggio in reti di petri
  - template tTransaction pTransaction delle ptpn

    - clock di uppaall
    - come si rappresenta uno stato nel model state graph di uppaal
    - JSemaphore
    - Parametro Lambda delle simulazioni ad attori

Sistemi Distribuiti e Cloud Computing (6 CFU e 9 CFU)

Talia Domenico

2018 2019

- Aloeasy
    - Java Card
    - Replicazione
    - NFS
    - COnsistenza

2019 2020

- Giovanni Giordano
    - Weak Consistency
    - release consistency
    - differenze EC2, S3 e DNS
- Anonimi
    - eukaliptus
    - Naming in generale
    - HT Condor

2020 2021

- Anonimi
    - componenti del Cloud Amazon
    - tecniche di scalabilità dei sistemi distribuiti
    - grid computing
    - Consistenza debole (synchronize)
    - Naming in generale e p2p
    - Kerberos
    - grid
    - algoritmo elezioni
- Erma_TV
    - HTCondor
    - Client Side Consistency (Eventual Consistency)
    - RPC (in prticolare RPC one-way)
    - Eucalyptus

2021 2022

- Anonimi
- prima sessione di interrogazione:
- ClassAds di HTCondor
    - cos'è e come viene usato il KDC
    - algoritmi di elezione
    - Eucalyptus
    - Match macker (ht condor)
    - Locking nfs
    - Naming sistemi distribuiti
    - seconda sessione di interrogazione:

        - MPI
        - Modello di autenticazione challenge-response a 5 messaggi a 3 e reflection Attack
        - File locking in NFS
        - sistemi distribuiti in generale e proprietà
        - Coda
        - Needham Shroeder
        - Kdc
        - RPC
        - Globus Gram Home based
        - mutua esclusione
        - NFS
        - lamport
        - sincronizzazione
        - Htcondor
        - consistenza sequenziale
        - read your writes
    - terza sessione di interrogazione:
        - Naming
        - Consistenza
    - quarta sessione di interrogazione:
        - Sistemi grid
        - HT condor
        - KDC
        - NFS lock
        - Strong mobility

Loris Belcastro

2018 2019

- Aloeasy
    - Distribuited garbage collector
    - Storage di Azure
    - Fabric Controller di Azure
    - come si passano i parametri in JAva RMI

2019 2020

- Giovanni Giordano
    - distribuited garbage collector
    - riferimenti Java RMI
    - tabelle Azure
    - Combiner

2020 2021

- Anonimi
    - equals in RMI
    - distributed garbage collector
    - tables di azure
    - json web token
    - Dynamic class download
    - Oggetti attivabili
    - Modulo combiner in map reduce
    - combiner
    - jwt
    - gerarchia row timestamp
- Erma_TV
    - MapReduce

    - Distributed Garbage Collector
    - Tables Di Azure

2021 2022

- Anonimi
  - prima sessione di interrogazione:
  - Map Reduce
  - La table di azure
  - dynamic class download
  - problema dell'equals in RMI e Remote Object
  - CDN
  - Combiner di MapReduce
- seconda sessione di interrogazione:
  DGB
  - garbage collector
  - storage di Azure
  - Docker in generale
  - come aggiungere un altro layer ad un'immagine
  - il vantaggio dei volumi sui bind mount
  - se esistono container con kernel Windows
  - differenze tra storage per oggetti e blocchi in aws
  - terza sessione di interrogazione:
    - Map reduce con Disegno e spiegazione del Partitioner e Combiner
    - Garbage collector
    - Table di azure
    - CND azure

Basi di Dati evolute

Molinaro Cristian

2019 2020

- Rak
  - calcolo relazionale e definizione di linguaggio indipendente dal dominio di valutazione
  - lock su database distribuiti
    - tecniche di assegnazione
    - deadlock
      - risposta: che se due transazioni richiedono il lock in scrittura sulla stessa risorsa e ci sono dei ritardi nella rete, nessuna delle due transazioni ottiene il lock e quindi si va in deadlock

2020 2021

- Anonimi
  - protocollo zero knowledge
  - algoritmo fiat shamir
  - proprietà funzioen hash firma digitale
  - paradosso compleanno

Calcolo Numerico

Yaroslav Sergeyev

2019 2020

- Anonimi
    - equazioni differenziali metodi conosciuti impliciti ed esplici
    - esistenza polinomio di interpolazione e tecniche con vantaggi e svantaggi (LaGrange e Newton)
    - metodo romberg
    - metodi Runge Kutta
    - metodi di interpolazione conosciuti (LaGrange ecc)
    - punto fisso condizioni convergenza
    - grafici di convergenza
    - metodi di derivazione numerica

2020 2021

- Anonimi
    - le tecniche di preprocessamento dei sistemi lineari (pivoting parziale, totale e bilanciamento)
    - indice di condizionamento
- Erma_Tv
    - integrale di riferimento
    - metodi di integrazione in più dimensione e perché non si può sempre suddividere in somma di integrali come in 1 dimensione
    - condizione convergenza metodi iterativi (sistemi)
    - ordine dell'errore (sia locale che globale) in tutti i metodi sulla risoluzione delle equazioni differenziali
    - può succedere che Jacobi converga e Gaus-Siedel diverga o viceversa?
    - FARE BENE il metodo di Cavalieri-Simpson (con enfasi sul motivo per cui si fa l'ipotesi sull' uguaglianza tra la derivata in psi e psi con tilde
    - come scegliere i nodi per evitare fenomeno Runge
    - modo migliore per calcolare la somma di tanti numeri in virgola mobile (slide Marat)
    - come si migliora l'indice di condizionamento? -> PREPROCESSING
- Anonimi
    - Quando parliamo di integrazione, cos'è l'intervallo di riferimento?
    - Qual'è il significato del condizionamento di un sistema lineare?
    - Cos'è la fattorizzazione di Cholesky?
    - Qual'è la differenza tra errore locale e errore globale
    - Qual'è il grado più elevato che permette di usare un polinomio di interpolazione?
        - (Risposta: settimo, oltre avviene il fenomeno di Runge)
    - Qual'è la migliore predisposizione dei nodi?
        - (Risposta: la peggiore sono i nodi equidistanti, la migliore sono i nodi di Chebyshev)
    - Vantaggi e svantaggi di metodi iterativi rispetto ai metodi diretti
        - (Risposta: sono più semplici ma non è detto che convergano)
    - Da cosa dipende il condizionamento di un sistema lineare?
    - Cancellazione numerica e come si può evitare
    - Prendendo un metodo iterativo qual'è la condizione della convergenza?
        - (Raggio spettrale (ovvero massimo degli autovalori della matrice d'iterazione) < 1)
    - Cos'è uno spazio lineare?
    - Data una grande sequenza di numeri positivi, qual'è il migliore modo di sommarli?
        - (Risposta: ordine crescente, minor perdita d'informazioni)
    - Quale dei metodi (Gauss e Gauss-Jordan) è il più efficente?
      Risposta: Il migliore è il metodo di Gauss perché ha una complessità minore
    - Svantaggi della formula del polinomio interpolante di LaGrange?
      Risposta: la complessità e non si possono aggiungere nodi senza dover ricalcolare il polinomio da capo
    - Significato di errore assoluto e relativo nell'approssimazione di un numero floating point
    - Formula adattiva di Cavalieri-Simpson e qual'è il presupposto fatto?
      Risposta: la derivata quarta di f(xi) è supposta uguale all'aumentare del passo
    - Quali sono i metodi per la risoluzione di equazioni differenziali ordinarie? Cosa vuol dire implicito ed esplicito?

- Residuo dei sistemi lineare? Se il residuo è piccolo cosa possiamo dire sulla soluzione?
  - Risposta: r^(k) = b - Ax^(k)
- Se il sistema è mal condizionato il fatto che il residuo è piccolo non ci dice nulla
- Metodi per la risoluzione di equazioni differenziali e ordine degli errori
- Come funzionano i metodi di integrazione numerica in più dimensioni? Perchè non si può usare la formula che trasforma un
- integrale a più dimensioni in una successione di integrali in una dimensione?
- Metodi per la derivazione numerica
- Estrapolazione di Richardson
- Migliorare il condizionamento di un sistema lineare?
  - Risposta: tecniche di pre-processing
- Metodi iterativi per la risoluzione dei sistemi lineari? Differenza in implementazione?
  - Risposta: Jacobi può essere parallelizzato
- Cos'è una matrice di permutazione e quali sono le proprietà?
- Formula di Cavaglieri-Simpson adattiva e come si valuta l'errore
- Fenomeno Runge e come si risolve?
  - Risposta: nodi di Chebyshev o uso di Spline
- Può capitare che uno dei metodi di risoluzione dei sistemi lineari (iterativi) converge e l'altro diverge?
  - Risposta: si perché avendo la matrice di iterazione due formule diverse il raggio spettrale potrebbe essere diverso
- Teorema dell'esistenza di un unico polinomio d'interpolazione
- Vantaggi e svantaggi dei metodi diretti rispetto ai metodi iterativi per la soluzione di sistemi lineari.
- Quando i metodi diretti non sono applicabili?
  - Risposta: Quando le matrici sono di grandi dimensioni è preferibile usare il metodo di Jacobi che è parallelizzabile
- Metodo dei coefficenti indeterminati?
- Metodo del punto fisso
- Condizione di Lipshiz e dove si applica
- Tipi di problemi computazionali (problema diretto, inverso e di indentificazione) ed esempi
- Pre-processing sistemi lineari
- polinomi osculatori
- spazi lineari
- metodo dei coefficienti indeterminati
- classificazione problemi computazionali
- integrazione in multi dimensioni
- CONDIZIONE DI LIPSCHITZ
- gauss e gauss jordan
- come trovare la matrice inversa
- matrice di permutazione
- qual è il trucco dea formula di integrazione di cavalieri Simpson?
- metodi di derivazione,i tipi e qual è il margine di errore, come si migliora, che grado di errore c'è
- clark nicolson
- calcolo delle matrici LU (con studio dell'errore)

Marat Mukhametzhanov

2019 2020

- Giovanni Giordano
    - errore assoluto e relativo
    - estrapolazione di Richardson
- Anonimi
    - fenomeno Runge
    - cancellazione numerica
    - decomposizione triangolare con Teoremi

2020 2021

- Anonimi
    - estrapolazione di richardson
    - Problema di Cauchy
    - Equazione differenziale
    - Stima indice K(A)
    - Differenze divise e proprietà
    - idea di fondo degli algoritmi
        - jacobi
    - Spline lineari e quadratiche

Algoritmi di Crittografia

Cristian Molinaro

2019 2020

- Giovanni Giordano
    - CBC
    - funzioni hash
- Anonimi
    - merkel puzzle
        - obiettivo
        - problemi
        - algoritmo
    - One Time Pad
        - decifatura e cifratura deterministica
        - decifatura e cifratura randomizzata
        - sicurezza per mandare messaggi
        - problemi
    - sicurezza Semantica
    - probab adv dice 1 quando EXP1
    - modi operativi many time Key
    - PRG e definizioni sicurezza
    - firma digitale e CA

2020 2021

- Anonimi
    - Modi operativi many time key
    - Sicurezza modi operativi many time key
    - zero knowledge
    - Algoritmo che è capace di attaccare qualsiasi funzione hash e paradosso del compleanno