Indice delle domande degli esami orali: Ingegneria Informatica LM

Questo file contiene le testimonianze degli esami orali di vari studenti del corso di laurea di **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!

- Indice delle domande degli esami orali: Ingegneria Informatica LM
- Architetture e programmazione dei sistemi di elaborazione
  - Fabrizio Angiulli
- Crittografia e analisi reti sociali
  - Molinaro Cristian
- Linguaggi Formali
  - Domenico Saccà
  - Rullo
- Informatica teorica
  - Scarcello Francesco
- Ottimizzazione
  - Maria Flavia Monaco
- Valutazione delle prestazioni
  - Pasquale Legato
- Intelligenza Artificiale (6 CFU)
  - Palopoli Luigi
- Intelligenza Artificiale e rappresentazione della conoscenza (12 CFU)
  - Palopoli Luigi
- Sistemi Informativi
  - Cassavia
- ISSTRA Ingegneria del software per sistemi real-time ed agenti
  - Libero Nigro
- 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

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

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

[201 201]

- 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

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`

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

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

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

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

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

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

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

Marat Mukhametzhanov

2019 2020

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

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