

**SISTEMI DISTRIBUITI E CLOUD COMPUTING A.A. 2012/13**  
**Prima prova intermedia - 10/12/2012**

**Cognome** \_\_\_\_\_ **Nome** \_\_\_\_\_

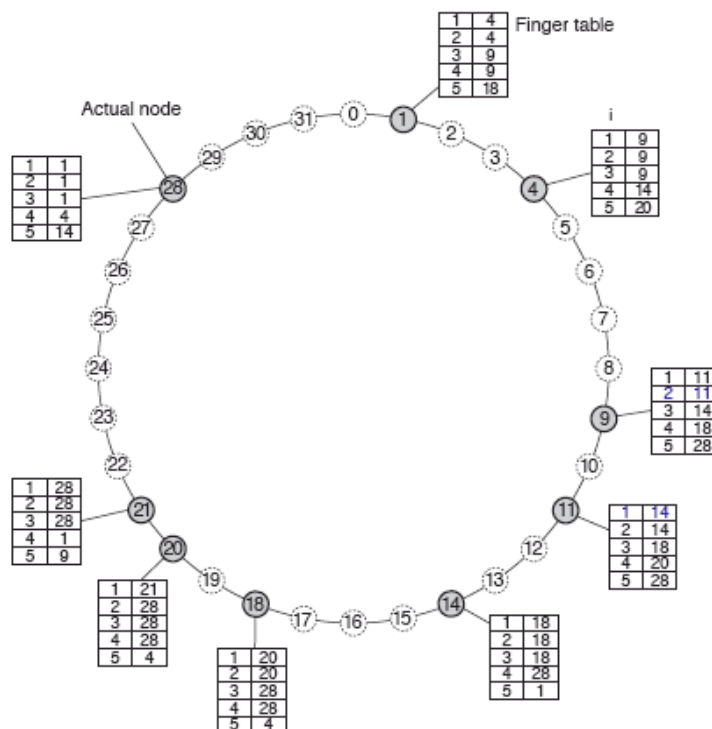
**Matricola** \_\_\_\_\_

**Domanda 1 (punti 5)**

Si spieghi il funzionamento di un'invocazione di metodo remoto (RMI). RMI supporta una forma di comunicazione transiente e sincrona: si spieghi cosa significa e quali sono i principali svantaggi. Inoltre, si presenti una possibile soluzione per rendere trasparente rispetto all'accesso, alla replicazione ed ai guasti l'invocazione di un metodo remoto replicato su molteplici server.

**Domanda 2 (punti 6)**

Si descrivano le caratteristiche, i vantaggi e gli svantaggi dei sistemi P2P basati su hash table distribuite. Con riferimento al sistema Chord basato su DHT raffigurato nella figura sottostante, si spieghino i valori della finger table del nodo 9 e si spieghi come viene effettuata da parte del nodo 4 la ricerca per la chiave 29.



**Domanda 3 (punti 6)**

Si descriva l'algoritmo di Maekawa per la mutua esclusione distribuita, evidenziandone le differenze rispetto agli algoritmi basati su autorizzazione esaminati a lezione. Quali proprietà relativamente a safety e liveness sono garantite da questo algoritmo? Infine, si confronti il suo costo in termini di comunicazione per ogni accesso alla sezione critica rispetto all'algoritmo di Ricart-Agrawala.

**Domanda 4 (punti 7)**

Si consideri il seguente problema di sincronizzazione in cui molteplici thread usano un buffer contenente un singolo elemento di tipo `char`. I thread possono inserire un elemento nel buffer oppure estrarre un elemento presente nel buffer. L'operazione `void put(char elem)` inserisce l'elemento nel buffer; se il buffer è pieno, il thread produttore chiamante viene sospeso in attesa che il buffer venga svuotato. L'operazione `char get(void)` permette di estrarre l'elemento dal buffer; se il buffer è vuoto, il thread consumatore chiamante viene sospeso in attesa che il buffer venga riempito da un altro thread. Si richiede che la gestione dell'attesa dei thread produttori sia di tipo FIFO, ovvero che venga risvegliato il thread che ha invocato da più tempo l'operazione di inserimento.

Si presenti lo pseudocodice in stile C di una soluzione per questo problema, spiegando quali funzioni e costrutti dell'API Pthreads esaminati durante il corso possono essere usati per risolvere il problema.

**Domanda 5 (punti 6)**

Si descrivano le caratteristiche della virtualizzazione completa e della paravirtualizzazione, confrontando i due approcci ed evidenziandone vantaggi e svantaggi. Si descriva come è possibile effettuare la migrazione live di macchine virtuali e quali vantaggi comporta per un provider di servizi IaaS.