

Prima parte

Esercizio 1 (fino a 5 punti)

Si descriva, anche mediante figure opportunamente commentate, quali sono i principi che Chord utilizza per memorizzare e per cercare una risorsa sulla rete.

Esercizio 2 (fino a 5 punti)

Si descrivano gli attacchi su rete di tipo DOS, SYN flooding e smurfing.

Esercizio 3 (fino a 5 punti)

Si realizzi un Web Service che permette di ottenere alcune informazioni sugli esami sostenuti dagli studenti di un corso di laurea. In particolare, il servizio espone:

1. un metodo che, data una *matricola*, restituisce l'elenco degli esami sostenuti da quello studente (per ogni esame viene restituito il CODICE dell'esame).
2. un metodo che, dati il codice di un esame (String) e un voto, restituisce la lista degli studenti che hanno conseguito un voto maggiore o uguale in quell'esame.

come specificato nel file WSDL allegato. Si implementi in Java una classe che implementa il servizio. Scegliere le opportune strutture che: i) permettono di salvare i prodotti venduti; ii) ottimizzano l'esecuzione delle query.

Allegato all'esercizio 4

```
<wsdl:definitions
targetNamespace="http://www.examples.com/wsdl/EsamiService">
<wsdl:types>
<schema targetNamespace="http://DefaultNamespace">
<import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
<complexType name="Studente">
  <sequence>
    <element name="Matricola" type="xsd:string"/>
    <element name="Nome" type="xsd:string"/>
    <element name="Cognome" type="xsd:string"/>
  </sequence>
</complexType>
<xs:complexType name="EsamiListType">
  <xs:sequence>
    <xs:element name="esami" type="xsd:string" minOccurs=0
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="StudentiListType">
  <xs:sequence>
    <xs:element name="studenti" type="tn1:Studente" minOccurs=0
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<schema> e
<wsdl:types>
  <wsdl:message name="EsamiStudenteRequest">
```

```
<wsdl:part name="in0" type="xsd:string"/>
</message>
<wsdl:message name="EsamiStudianteResponse">
  <wsdl:part name="in0" type="tns1:EsamiListType"/>
</message>
<wsdl:message name="StudentiPerEsameRequest">
  <wsdl:part name="in0" type="xsd:string"/>
  <wsdl:part name="in1" type="xsd:integer"/>
</message>
<wsdl:message name="StudentiPerEsameResponse">
  <wsdl:part name="in0" type="tns1:StudentiListType"/>
</message>
</wsdl:types>

<wsdl:portType name="EsamiService">
<wsdl:operation name="EsamiStudiante" parameterOrder="in0">
  <wsdl:input message="impl:EsamiStudianteRequest"
name=EsamiStudianteRequest/>
  <wsdl:output message="impl:EsamiStudianteResponse"
name=EsamiStudianteResponse/>
</operation>
<wsdl:operation name="StudentiPerEsame" parameterOrder="in0 in1">
  <wsdl:input message="impl:StudentiPerEsameRequest"
name=StudentiPerEsameRequest />
  <wsdl:output message="impl:StudentiPerEsameResponse" name=
StudentiPerEsameResponse />
</operation>
</wsdl:portType>

<wsdl:binding ...>
...
</wsdl:binding>
<wsdl:service ...>
...
</wsdl:service>
</wsdl:definitions>
```