

### Esercizio 1

Dato il seguente schema relazionale, si individui in ISCRIZIONE il *fatto* di interesse, si costruisca l'*attribute tree* e il *fact chema* e si definisca la struttura relazionale dello *star schema*.

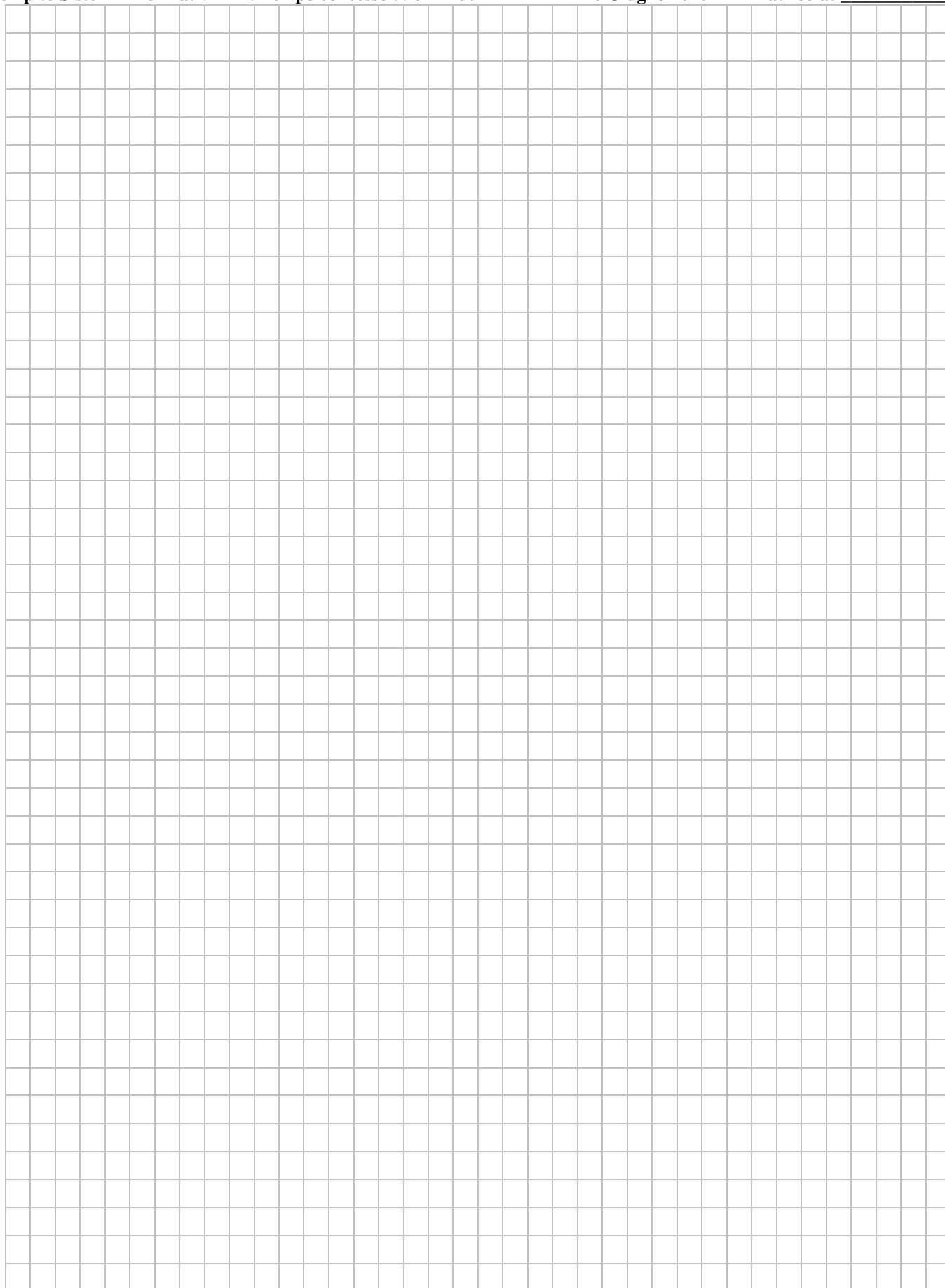
**LEZIONE**(IDLezione, Data, Ora, corso:CORSO, Durata);  
**FREQUENZA**(IDLezione:LEZIONE, Data:LEZIONE, Ora:LEZIONE, CF:STUDENTE);  
**CORSO**(Nome, TipoCorso, LivelloDifficoltà, AnnoCorso);  
**ISCRIZIONE**(CF:STUDENTE, IDCorso:CORSO);  
**PROGRAMMA**(Corso:CORSO, Argomento, Descrizione);  
**STUDENTE**(CF, Cognome, Nome, Indirizzo, Citta, Regione, Stato, Telefono);

Utilizzando il formalismo delle *espressioni dimensionali* formulare le seguenti interrogazioni:

1. Numero totale di iscritti a corsi di tipo “informatica”, per ciascun anno
2. Corsi per cui il numero totale di iscritti sia maggiore di 50

### Svolgimento

A large grid of graph paper for writing the solution, consisting of many small squares.



**Esercizio 2**

Date le relazioni:

FORNITORI( pi, nome, indirizzo, nazione )  
 PARTI( cod, piFornitore:FORNITORI, colore, prezzo )

e la query:

```
SELECT FORNITORI.nome, PARTI.prezzo
FROM FORNITORI, PARTI
WHERE FORNITORI.pi = PARTI.piFornitore
AND PARTI.colore = 'Rosso'
AND FORNITORI.nazione = 'Belgio'
```

Calcolare il *numero atteso* di tuple del risultato della query e il *miglior piano d'accesso* fra i seguenti:

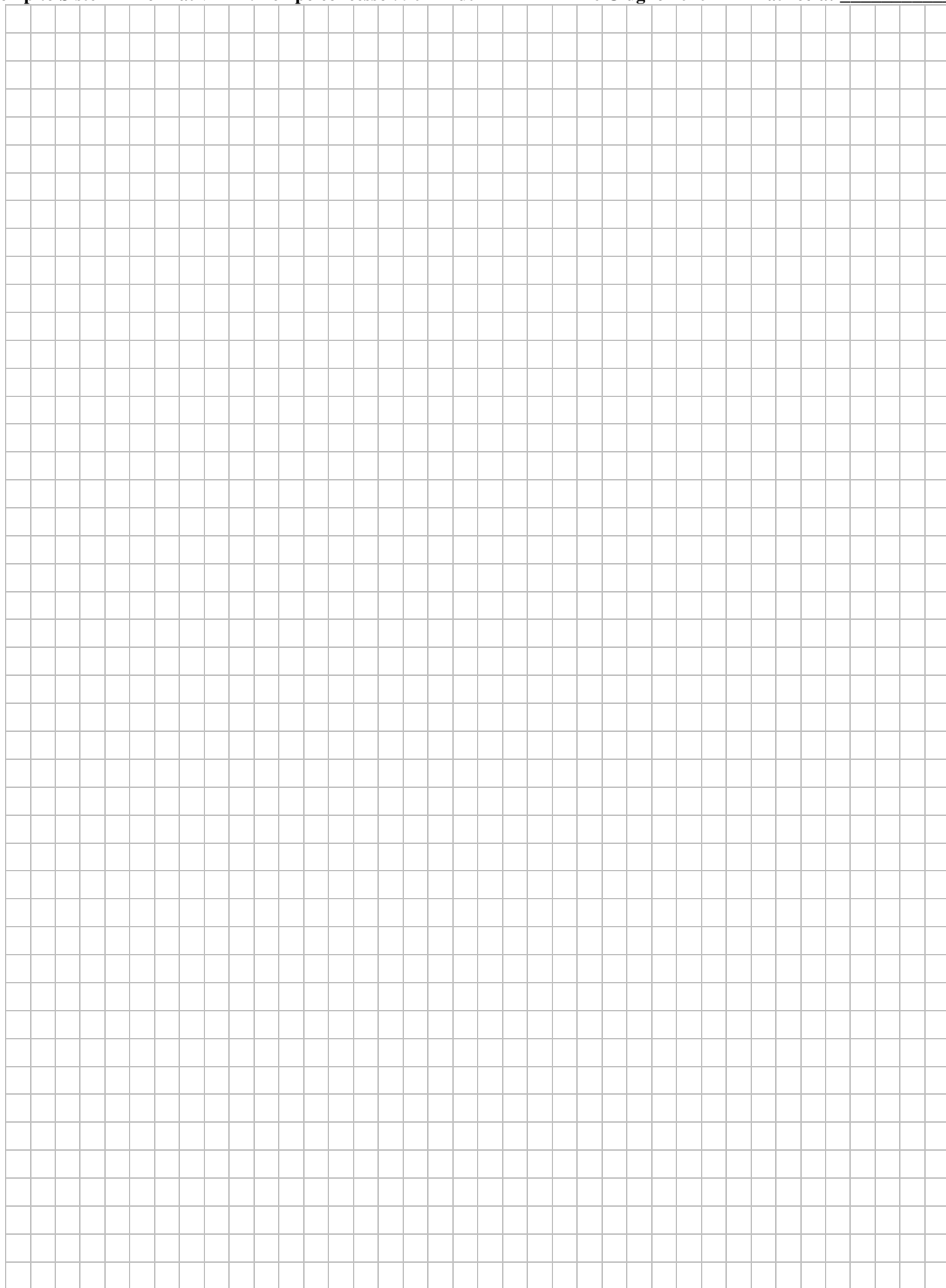
1. nested-loop con relazione interna PARTI
2. nested-loop con relazione interna FORNITORI
3. sort-merge a Z=3 vie

nell'ipotesi che:

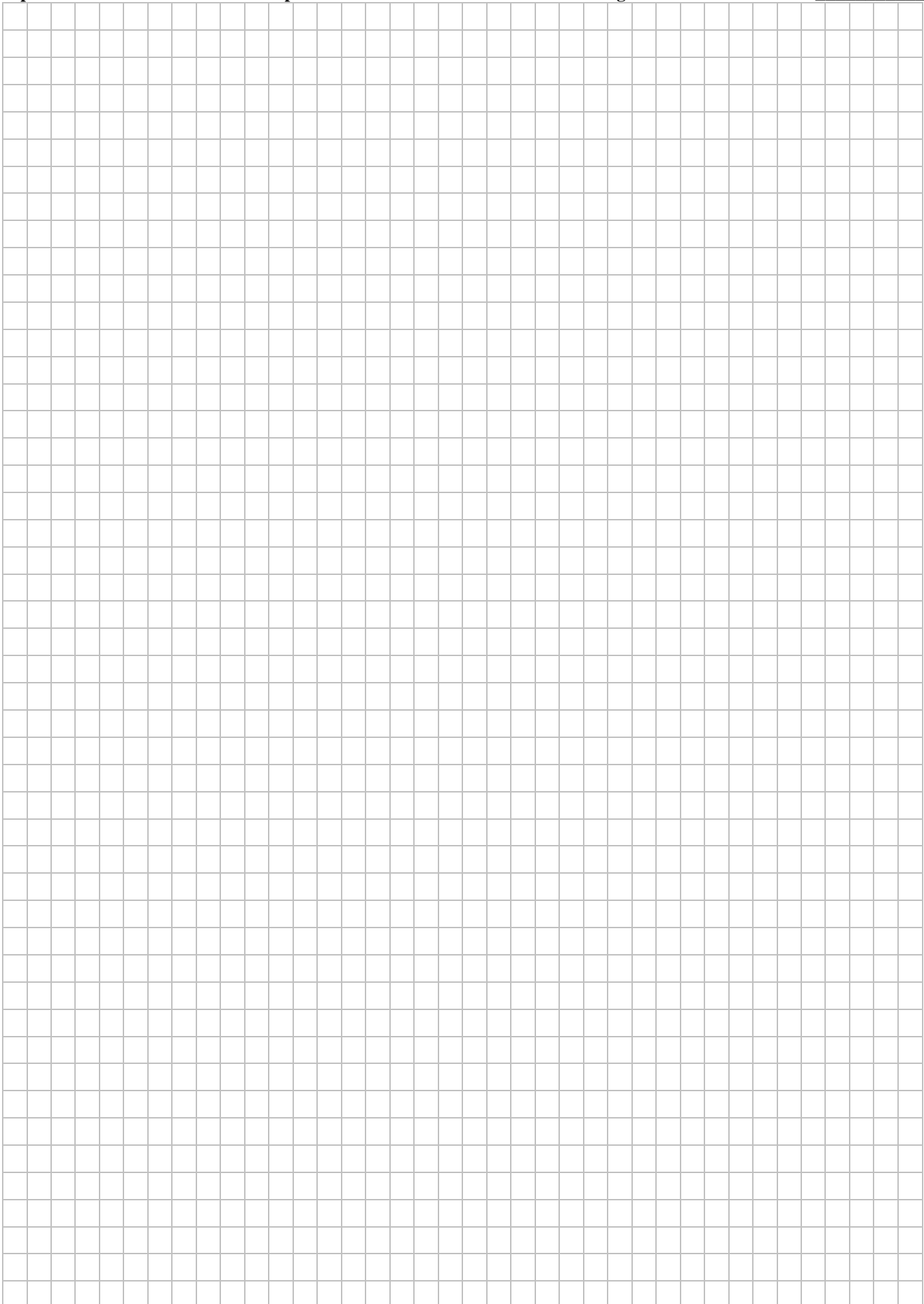
$NP_{FOR} = 160$                        $NT_{FOR} = 5.000$   
 $NP_{PAR} = 550$                        $NT_{PAR} = 150.000$   
 $NK_{colore} = 10$                        $NK_{nazione} = 40$   
 $len(cod)=4$                        $len(pi)=4$                        $len(prezzo)=4$                        $len(colore)=20$

e sapendo che su FORNITORI.pi è costruito un indice clustered con  $NL_{pi} = 10$  e  $h = 2$

**Svolgimento**







**Esercizio 4**

E' dato un file di record organizzato secondo la tecnica del Virtual hashing composto inizialmente di  $NP_0$  bucket ( $NP_0=4$ ) di capacità C ( $C=3$ ). La situazione è la seguente:

0	116	1
	144	
1	225	1
	37	1
	657	1
	722	
3	15	
	99	
	935	

**V**

- 1) Mostrare l'evoluzione del file dopo l'inserimento del record di chiave 487 calcolando i costi di accesso in termini di numero di operazioni di I/O.
- 2) Spiegare la differenza tra strutture hash di tipo statico e dinamico elencando qualche esempio per ciascuna classe.

**Svolgimento**

**FORMULE DI RIEPILOGO PER IL CALCOLO DEI COSTI**

- **Dimensione di una relazione:**  $NP = \lceil NR \times \text{len}(t) / (D \times u) \rceil$
- **Costo di ordinamento (Sort Merg Z vie):**  $2 \times NP \times \lceil \log_2 NP \rceil$
- **Scansione sequenziale tabelle**
  - in media a  $(NP+1)/2$  blocchi
  - nel caso peggiore a NP blocchi
  - in caso di non esistenza del record si accede a NP blocchi
- **Scansione binaria tabelle**
  - in media a  $\lfloor \log_2 NP \rfloor$  blocchi
  - nel caso peggiore a  $\lfloor \log_2 NP \rfloor + 1$  blocchi
  - In caso di non esistenza del record si accede a  $\lfloor \log_2 NP \rfloor + 1$  blocchi
- **Accesso a tabelle con indice**
  - **Numero di foglie del B+-Tree:**  $NL = \lceil (NK \cdot \text{len}(k) + NR \cdot \text{len}(p)) / (D \cdot u) \rceil$
  - **altezza del B+-Tree:**  $1 + \left\lceil \log_{g+1} \left( \frac{NN}{2} \right) \right\rceil, NN = \min(NL, NK)$
  - **ordine del B+-Tree:**  $g = \left\lfloor \frac{D - \text{len}(p)}{2(\text{len}(k) + \text{len}(p))} \right\rfloor$
  - **Indice Clustered:**  $h - 1 + \lceil EK / NK \cdot NL \rceil + \lceil EK / NK \cdot NP \rceil$
  - **Indice Unclustered:**  $h - 1 + \lceil EK / NK \cdot NL \rceil + EK \cdot \Phi(NR / NK, NP)$
  - **Formula di Cardenas**  $\Phi(ER, NP) = NP \times \left( 1 - \frac{1}{NP} \right)^{ER} \leq \min ER, NP$
- **Proiezione**
  - **Mantenendo la chiave della relazione:** NP
  - **Perdendo la chiave della relazione:** assimilabile al costo di ordinamento
- **Raggruppamento: assimilabile al costo di ordinamento**
- **Nested loop:**
  - **senza predicato di selezione**  $NP_R + NR_R \times NP_S$
  - **con predicato di selezione**  $C_a(R) + (\text{sel}(\text{pred}) \times NR_R) \times C_a(S)$
- **Sort merge**
  - **Tabelle già ordinate**  $NP_R + NP_S$
  - **Tabelle da ordinare**  $\text{Sort}(R) + \text{Sort}(S) + NP_R + NP_S$
- **Fattori di selettività**
  - **Predicato "="**  $f(A=v) = 1/NK_A$
  - **Predicato "IN"**  $f(A \in \text{set}) = \text{card}(\text{set})/NK_A$
  - **Predicato "<"**  $f(A < v) = \frac{v - \min(A)}{\max(A) - \min(A)} \times \frac{NK_A - 1}{NK_A}$
  - **per attributi con molti valori:**  $f(A < v) = \frac{v - \min(A)}{\max(A) - \min(A)}$
  - **Predicato "between"**  $f(A \in [v1, v2]) = \frac{v2 - v1}{\max(A) - \min(A)} \times \frac{NK_A - 1}{NK_A} + \frac{1}{NK_A}$
  - **per attributi con molti valori:**  $f(A \in [v1, v2]) = \frac{v2 - v1}{\max(A) - \min(A)}$
  - **f(p1 AND p2)**  $= f_{p1} \times f_{p2}$
  - **f(p1 OR p2)**  $= f_{p1} + f_{p2} - f_{p1} \times f_{p2}$