



Esercizi di riepilogo sui dispositivi e sulle organizzazioni dei dati

Dario Maio

<http://bias.csr.unibo.it/maio/>



Esercizio n° 1

- È dato un file memorizzato su disco in NP blocchi; ogni blocco è accessibile in modo diretto specificandone l'indirizzo logico ind (da 1 a NP); dato il seguente frammento di codice:

```
if odd(NP)
then npassi:= NP div 2
else npassi:= NP div 2 - 1;
for p:=1 to npassi do
begin
    ind:=p+1;
    for j:=1 to npassi-p+1 do
    begin
        scambia_blocco(ind,ind+1);
        ind:=ind+2
    end
end;
```

dove `scambia_blocco(j,k)` è una procedura che scambia il contenuto del blocco di indirizzo logico j con il contenuto del blocco di indirizzo logico k.

- Si determini l'effetto prodotto dal codice sul file e si valuti, in funzione di NP, il numero di operazioni di lettura e scrittura su disco.
- Si assuma per ipotesi di disporre di due buffer in memoria centrale, ciascuno atto a ospitare un singolo blocco, il primo **I/O_BUFF** da adibire alle operazioni di lettura e scrittura, il secondo **AUX_BUFF** da utilizzare come appoggio per lo scambio.



Esercizio n° 1 - Soluzione (1)

- L'insieme dei blocchi può essere pensato come partizionato in due sottoinsiemi: il sottoinsieme dei blocchi di indirizzo dispari e il sottoinsieme dei blocchi di indirizzo pari.
- L'esecuzione del codice produce come effetto di compattare all'inizio tutti i blocchi che avevano indirizzo dispari e in fondo quelli che avevano indirizzo pari, conservando l'ordine relativo degli indirizzi all'interno di ogni sottoinsieme.

Esempio:

prima dell'esecuzione

1	2	3	4	5	6	7	indirizzo
a	b	c	d	e	f	g	contenuto

dopo l'esecuzione

1	2	3	4	5	6	7	indirizzo
a	c	e	g	b	d	f	contenuto

```

if odd(NP)
then npassi:= NP div 2
else npassi:= NP div 2 - 1;
for p:=1 to npassi do
begin
ind:=p+1;
for j:=1 to npassi-p+1 do
begin
scambia_blocco(ind,ind+1);
ind:=ind+2
end
end;

```

Esercizio n° 1 - Soluzione (2)

- Si deriva facilmente che il numero di chiamate alla procedura `scambia_blocco` è esprimibile, in funzione di NP, come:

$$NC_{\text{pari}} = \sum_{i=1}^{\frac{NP}{2}-1} i = \frac{NP^2 - 2 \times NP}{8}, \text{ se NP è pari}$$

$$NC_{\text{dispari}} = \sum_{i=1}^{\frac{NP-1}{2}} i = \frac{NP^2 - 1}{8}, \text{ se NP è dispari}$$

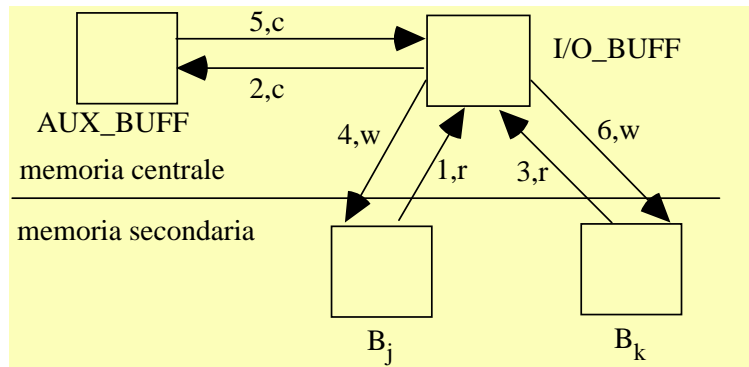
```

if odd(NP)
then npassi:= NP div 2
else npassi:= NP div 2 - 1;
for p:=1 to npassi do
begin
ind:=p+1;
for j:=1 to npassi-p+1 do
begin
scambia_blocco(ind,ind+1);
ind:=ind+2
end
end;
end;

```

Esercizio n° 1 - Soluzione (3)

- Si consideri ora la figura seguente, che illustra l'ordinamento temporale delle operazioni compiute per lo scambio di due generici blocchi B_j e B_k ; a fianco del numero d'ordine dell'operazione è indicato il rispettivo tipo (r = lettura di un blocco da disco, w = scrittura di un blocco su disco, c = copia di un blocco in memoria principale).



ogni scambio comporta:
2 letture da disco e
2 scritture su disco

per NP pari: $2 \times NC_{\text{pari}}$ letture ; $2 \times NC_{\text{pari}}$ scritture;

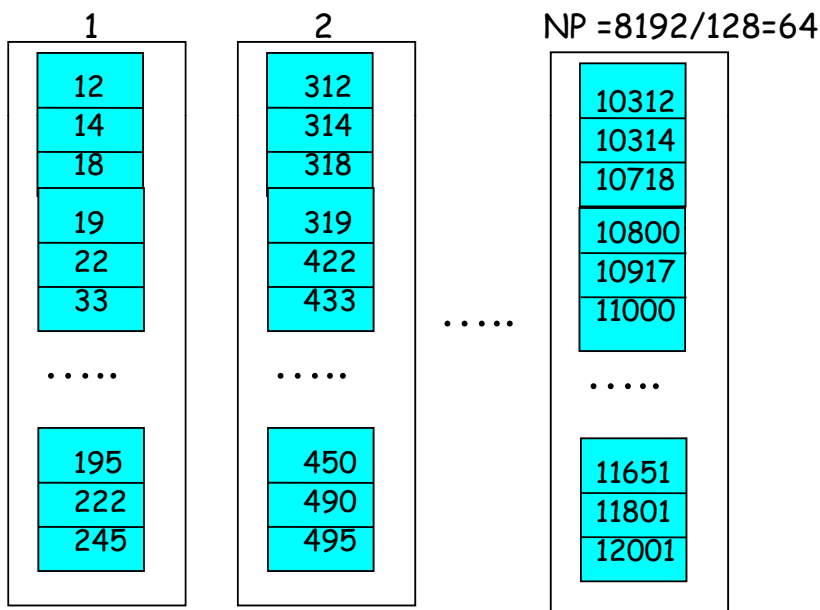
per NP dispari: $2 \times NC_{\text{dispari}}$ letture ; $2 \times NC_{\text{dispari}}$ scritture.

Esercizio n° 2

- Sono dati due insiemi di interi A di cardinalità M , e B di cardinalità N , memorizzati rispettivamente *in un array in memoria principale* e *in un file su disco*; il file è ordinato in senso crescente, l'array è disordinato. Sia $N=8192$ il numero di record nel file, memorizzati in blocchi di 128 record ciascuno, e sia $M=512$ il numero degli elementi dell'array.
- Si dispone di tre algoritmi Alg1, Alg2 e Alg3 per visualizzare gli interi facenti parte dell'insieme $A \cap B$, che operano rispettivamente secondo i criteri appresso descritti:
- Alg1) scandisce sequenzialmente il file e per ogni record ricerca se esiste già l'elemento nell'array;
- Alg2) scandisce sequenzialmente l'array e per ogni elemento applica una ricerca dicotomica sul file.
- Alg3) ordina dapprima l'array in memoria, quindi scandisce parallelamente l'array e il file.
- Supponendo che la complessità in tempo sia determinata principalmente dall'I/O su disco, si determini per ognuno dei tre algoritmi il caso peggiore e si calcoli il numero di accessi a blocchi di disco.

Esercizio n° 2 - Soluzione

Blocchi del file B ordinato



Array A disordinato

1	2222
2	422
	4312
	151
	18
	43
	4114
	10312
	1112
	1
	114
	21
.....	
512	312
	3120
	1140
	214

Esercizio n° 2 - Soluzione

Il file contiene $NP = 8192/128 = 64$ blocchi:

- Alg1) Nel caso peggiore accede a $NP = 64$ blocchi.
- Alg2) Per ogni elemento dell'array opera una ricerca dicotomica su file, pertanto, nel caso peggiore legge $M \times (\lfloor \log_2 NP \rfloor + 1) = 512 \times 7 = 3584$ blocchi.
- Alg3) Nel caso peggiore accede a $NP = 64$ blocchi.

Esercizio n° 3

- È dato un B-tree di ordine $g=9$ con $NK=999999$ chiavi. Si effettuano $M=10$ operazioni di cancellazione di chiave e ogni cancellazione avviene in una foglia lasciando un numero di chiavi residuo maggiore o uguale a g .
- Si stimi, nel caso peggiore, il costo complessivo delle M cancellazioni, in termini di numero di nodi letti e scritti, assumendo che la radice del B-tree resti disponibile in memoria principale dopo la prima cancellazione.

Esercizio n° 3 - Soluzione

L'algoritmo di cancellazione di un valore di chiave in un B-tree:

```
{ ricerca la chiave y;
  if trovata
  then { if y in una foglia L
        then cancella y da L
        else {ricerca nel sottoalbero di destra fino a una foglia L seguendo la
              catena dei puntatori;
              sostituisci a y la prima chiave di L;
              cancella la prima chiave di L
            };
        if L contiene meno di g chiavi
        then attiva catenation e underflow
      }
}
```

Esercizio n° 3 - Soluzione

- Per la prima cancellazione si leggono h nodi e si riscrive un nodo, per ogni altra cancellazione si leggono $h-1$ nodi e si riscrive un nodo.
- Dunque si operano $M \times (h - 1) + 1$ letture e M scritture.
- Si ricorda che se il B-tree contiene NK chiavi vale la seguente relazione:

$$\lceil \log_{2g+1}(NK + 1) \rceil \leq h \leq 1 + \left\lfloor \log_{g+1} \frac{NK + 1}{2} \right\rfloor$$

- Con i dati del problema si deriva che $5 \leq h \leq 6$. Per il caso peggiore si fa riferimento all'altezza massima $h=6$ e quindi in totale si leggono 51 nodi e si riscrivono 10 nodi.

Esercizio n° 4

- È dato un file con NR record memorizzati in NP pagine. Sul j -mo campo è costruito un **secondary unclustered B⁺-tree a PID**; siano rispettivamente NK_j il numero di valori distinti del campo j e DK_j la cardinalità del dominio del campo j .
- Si calcoli la probabilità di aggiornare l'indice (ovvero aggiungere un nuovo PID) inserendo un nuovo record nel file, sapendo che il record viene inserito in una delle NP pagine già esistenti.

Esercizio n° 4 - Soluzione

- Due sono le situazioni in cui, a seguito dell' inserimento di un nuovo record nel file dati, è necessario aggiungere un nuovo PID nell'indice:
 - ❖ il valore del campo j del nuovo record non è già presente nel file e quindi nell'indice stesso;
 - ❖ il valore del campo j del nuovo record, pur già presente nel file, non è presente nella pagina dati dove il record viene inserito.
- Con ipotesi di uniformità e supponendo che ogni pagina contenga mediamente un ugual numero di record pari a NR/NP , la probabilità di aggiornare l'indice a causa di un inserimento di un nuovo record può essere calcolata come:

$$\left(1 - \frac{NK_j}{DK_j}\right) + \frac{NK_j}{DK_j} \times \left(1 - \frac{1}{NK_j}\right)^{\frac{NR}{NP}}$$



Esercizio n° 5

- Calcolare l'occupazione di memoria di un secondary B⁺-tree con i seguenti dati:

dimensione di un nodo: $D = 1\text{KB}$
lunghezza puntatore: $\text{len}(p) = 4 \text{ byte}$
lunghezza chiave: $\text{len}(k) = 10 \text{ byte}$
numero record: $NR = 1000000$
numero chiavi: $NK = 1000$
utilizzo: $u = \ln 2$



Esercizio n° 5 - Soluzione

- Trascurando per semplicità la presenza del puntatore alla foglia successiva, si determina il numero delle foglie come:

$$NL = \left\lceil \frac{NK \times \text{len}(k) + NR \times \text{len}(p)}{D \times u} \right\rceil = 5650$$

- Assumendo che la lunghezza di un puntatore a un nodo intermedio richieda $\text{len}(q) = \text{len}(p)$ byte, è possibile derivare l'ordine g del B⁺-tree, utilizzando la relazione:

$$2g \times \text{len}(k) + (2g + 1) \times \text{len}(q) \leq D$$



Esercizio n° 5 - Soluzione

- Dunque si ricava:

$$g = \left\lfloor \frac{D - \text{len}(q)}{2 \times (\text{len}(k) + \text{len}(q))} \right\rfloor = 36$$

- Per calcolare l'altezza dell'albero occorre considerare il fatto che, essendo $NL/NK \approx 5.6$, solo una foglia ogni cinque sarà indirizzata al livello $h-1$. Sostituendo NK al posto di NL nella formula per calcolare l'altezza dell'albero si ottiene:

$$1 + \left\lceil \log_{2g+1} NK \right\rceil \leq h \leq 2 + \left\lfloor \log_{g+1} \frac{NK}{2} \right\rfloor$$

- e dunque $h_{\min} = h_{\max} = 3$.

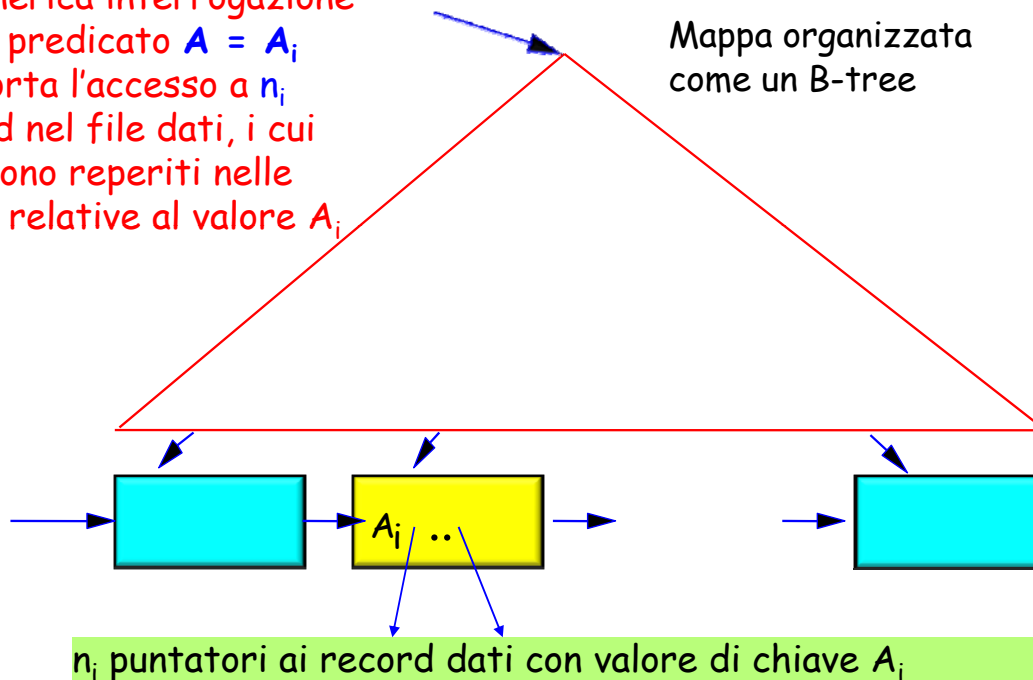


- ❖ 5650 blocchi a livello foglie
- ❖ $\left\lfloor \frac{NK}{g+1} \right\rfloor = \left\lfloor \frac{1000}{37} \right\rfloor = 27$ a livello 2
- ❖ 1 radice

- Il numero di valori distinti di A è pari a NK_A e per ogni possibile valore A_i , $i=1,2, \dots, NK_A$, si conosce il numero n_i di record che presentano quel valore A_i nel file.
- È dato inoltre un insieme $Q=\{q_1, q_2, \dots, q_m\}$ di interrogazioni con predicato di selezione ($A = \text{valore}$) che sono risolte utilizzando l'indice suddetto come via di accesso.
- Si stimi il valor medio del costo globale di accesso a pagine dati e indice, necessario per risolvere tutte le m interrogazioni, sapendo che la probabilità p_i che una generica interrogazione q_j interessi il valore A_i è pari a n_i/NR .

Esercizio n° 6 - continua

La generica interrogazione q_j con predicato $A = A_i$ comporta l'accesso a n_i record nel file dati, i cui TID sono reperiti nelle foglie relative al valore A_i



Esercizio n° 6 - Soluzione

- La generica interrogazione q_j viene risolta via indice.
- Per l'accesso all'indice, poiché è sensato assumere che il singolo valore A_i comporti un numero di foglie pari a:

$$\left\lceil \frac{n_i}{NR} \times NL \right\rceil$$

quando q_j interessa il valore A_i si ha un costo:

$$C_I = h - 1 + \left\lceil \frac{n_i}{NR} \times NL \right\rceil$$

Esercizio n° 6 - Soluzione

■ Per quanto riguarda l'accesso ai dati si può ricorrere alla **formula di Cardenas** stimando per la generica interrogazione, quando richiede record con valore A_i dell'attributo A , un costo pari a:

$$C_D = \Phi(n_i, NP)$$

■ Allora il costo medio globale per l'insieme Q delle interrogazioni è:

$$m \times \sum_1^{NK_A} p_i \times (C_I + C_D)_i = m \times \sum_1^{NK_A} p_i \times \left(h - 1 + \left\lceil \frac{n_i}{NR} NL \right\rceil + \Phi(n_i, NP) \right)$$