Homework
Query Processing

Problem 1 *Compute.*

1. Let $M = 1000$, $B(R) = 5000$, $B(S) = 2000$. Determine the costs of
   (a) block nested-loops join $R \bowtie S$;
   (b) sort-based join $R \bowtie S$;
   (c) hash-based bag $R - S$;
   (d) sort-based $\delta(R)$;
   (e) sort-based $\delta(S)$;
   (f) index-based zig-zag join $R \bowtie S$;
   (g) hash-based set $R \cap S$;
   (h) sort-based $\gamma_L(R)$;
   (i) bag $R \cup S$;
   (j) hash-based set $R \cup S$.
2. Let $M = 200$, $B(R) = 5000$. Query processor determines that to compute $R \bowtie S$ using sort-based join algorithm it needs three passes (cannot do it in two passes). What is the smallest possible size of S (in terms of number of blocks)?
3. Let $B(R) = 700$. How many buffer slots (pages) do we need in order to compute $\gamma_L(R)$ using a two-pass hash-based algorithm?
4. Let $M = 20$, $B(R) = 40,000, B(S) = 5000$. How many passes will a sort-based algorithm for $B(R) \cap B(S)$ require?
5. Let $M = 50$, $B(R) = 10,000$, $B(S) = 40$. List all binary operations between R and S that can be performed in one pass.

6. Let $M = 100$, $B(R) = 50,000$. How small should $B(S)$ be to make is feasible to compute $B(R) - B(S)$ (set) in two passes? What algorithm would you use?

7. For each of the following values of M: $M \in \{20, 100, 500, 1000, 10,000\}$, and for two relations R and S of sizes $B(R) = 10,000$, $B(S) = 5,000$ determine the cheapest way to execute $R \bowtie S$.

Problem 2 *Consider the following database schema describing a league of basketball teams:*

```
CREATE TABLE Teams (
   Id     INT PRIMARY KEY,
   Name   CHAR(30),
   Coach  INT REFERENCES Coaches,
   Wins   INT,
   Losses INT,
   Place  INT
 );

CREATE TABLE Players (
   Id       INT PRIMARY KEY,
   Name     CHAR(30),
   Position CHAR(2),
   Height   INT,   /* in inches */
   Team     INT REFERENCES Teams
);
```

```
CREATE TABLE Coaches (
   Id    INT PRIMARY KEY,
   Team  INT REFERENCES Teams,
   Name  CHAR(30)
);

CREATE TABLE Games (
   Id            INT  PRIMARY KEY,
   HomeTeam      INT  REFERENCES Teams,
   HomeTeamScore INT,
   AwayTeam      INT  REFERENCES Teams,
   AwayTeamScore INT
 );
```

```
CREATE TABLE Stats (
  Player  INT REFERENCES Players,
  Game    INT REFERENCES Games,
  PTS     INT, /* points scored */
  AST     INT, /* assists */
  RB      INT, /* rebounds */
  BLK     INT, /* blocks */
  STL     INT, /* steals */
  TO      INT, /* turnovers */
  PF      INT, /* personal fouls */
  TF      INT, /* technical/flagrant fouls */
  PRIMARY KEY (Player, Game)
)
```

1. Let page size be 4K, block headers take 256 bytes, and records use padding to ensure that all constraints on field locations are satisfied. Describe the records for each of the tables. Deterimine how many records per page would the data files for each table store.

2. Using the results of part 1 of the problem, determine how many pages will be needed to store the given number of tuples in each file. Assume that the data file organization is heap, and that all pages but the last would store the largest number of records possible.

(a) $T(Teams) = 400$. Find $B(Teams)$.

(b) $T(Teams) = 275$. Find $B(Teams)$.

(c) $T(Teams) = 850$. Find $B(Teams)$.

(d) $T(Coaches) = 550$. Find $B(Coaches)$.

(e) $T(Coaches) = 2200$. Find $B(Coaches)$.

(f) $T(Coaches) = 3425$. Find $B(Coaches)$.

(g) $T(Players) = 5,000$. Find $B(Players)$.

(h) $T(Players) = 19,000$. Find $B(Players)$.

(i) $T(Players) = 32,000$. Find $B(Players)$.

(j) $T(Games) = 22,000$. Find $B(Games)$.

(k) $T(Games) = 40,000$. Find $B(Games)$.

(l) $T(Games) = 46,690$. Find $B(Games)$.

(m) $T(Stats) = 110,000$. Find $B(Stats)$.

(n) $T(Stats) = 400,000$. Find $B(Stats)$.

(o) $T(Stats) = 1,000,000$. Find $B(Stats)$.

3. Let:

Parameter	Value
M	500
$B(Teams)$	60
$B(Coaches)$	100
$B(Players)$	3,500
$B(Games)$	4,000
$B(Stats)$	10,500

Find the best algorithm to perform each of the following operations, compute the I/O costs:

(a) $\sigma_{Position=\text{"}PG\text{"}}(Players)$;

(b) $\pi_{Player,Game,AST}(Stats)$;

(c) $\sigma_{AST \geq 10}(Stats)$;

(d) $Players \bowtie_{Id=Player} Stats$;

(e) $Coaches \bowtie_{Id=Coach} Teams$;

(f) $Games \bowtie_{Hometeam=Id} Teams$;

(g) $\gamma_{Player,SUM(AST)}(Stats)$;

(h) $\gamma_{Hometeam,SUM(Homescore)}(Games)$;

(i) $\tau_{AST}(Stats)$;

(j) $\sigma_{Homescore>Awayscore}(Games)$;