Comp 543 Assignment 2

Kai-Po Lin

1.1 Connected Components

[Code]

```
-- 1.1 Connected Components
BEGIN
    DECLARE @tempVisited TABLE (
         id INT
    );
    INSERT INTO @tempVisited (id) (SELECT n.paperid FROM Nodes n);
    DECLARE @tempVisitCnt INT = (
         SELECT Count(*)
         FROM @tempVisited
    );
    WHILE (@tempVisitCnt > 0)
         BEGIN
             DECLARE @paperID INT, @connectedCnt INT=0;
             DECLARE @Visited TABLE (
                  id INT
             );
             DECLARE @NewAll TABLE (
                  id INT,
                  visCnt INT
             );
             DECLARE @Temp TABLE (
                  id INT,
                  visCnt INT
             );
             SET @paperID = (
                  SELECT TOP (1) v.id
                  FROM @tempVisited v
             );
             INSERT INTO @Visited VALUES (@paperid);
             UPDATE @NewAll SET visCnt = 0;
             INSERT INTO @Visited (id)
             OUTPUT inserted.id INTO @Temp(id)
```

```
SELECT DISTINCT e.citedPaperID
FROM Edges e
WHERE e.paperID = @paperID AND e.citedPaperID <> @paperID
UPDATE @Temp SET visCnt = 0;
UPDATE @NewAll SET visCnt += 1;
DELETE FROM @NewAll WHERE visCnt = 2;
INSERT INTO @NewAll (id, visCnt) (SELECT * FROM @Temp);
DELETE FROM @Temp;
INSERT INTO @Visited (id)
OUTPUT inserted.id INTO @Temp (id)
SELECT DISTINCT e.PaperID
FROM Edges e JOIN @Visited v ON e.citedPaperID = v.id
WHERE e.paperID NOT IN (SELECT v1.id FROM @Visited v1)
UPDATE @Temp SET visCnt = 0;
UPDATE @NewAll SET visCnt += 1;
DELETE FROM @NewAll WHERE visCnt = 2;
INSERT INTO @NewAll (id, visCnt) (SELECT * FROM @Temp);
DELETE FROM @Temp;
DECLARE @NewCnt INT = (SELECT COUNT(*) FROM @NewAll);
WHILE (@NewCnt > 0)
    BEGIN
        INSERT INTO @Visited (id)
        OUTPUT inserted.id INTO @Temp (id)
        SELECT DISTINCT e.citedPaperID
        FROM Edges e JOIN @NewAll a ON e.paperID = a.id
        WHERE e.citedPaperID NOT IN (SELECT v.id FROM @Visited v)
        UPDATE @Temp SET visCnt = 0;
        UPDATE @NewAll SET visCnt += 1;
        DELETE FROM @NewAll WHERE visCnt = 2;
        INSERT INTO @NewAll (id, visCnt) (SELECT * FROM @Temp);
        DELETE FROM @Temp;
        SET @NewCnt = (SELECT COUNT(*) FROM @NewAll);
        IF (@NewCnt > 0)
             BEGIN
                 INSERT INTO @Visited (id)
                 OUTPUT inserted.id INTO @Temp (id)
```

```
SELECT DISTINCT e.PaperID
                      FROM Edges e JOIN @NewAll a ON e.citedPaperID = a.id
                      WHERE e.paperID NOT IN (SELECT v.id FROM @Visited v)
                      UPDATE @Temp SET visCnt = 0;
                      UPDATE @NewAll SET visCnt += 1;
                      DELETE FROM @NewAll WHERE visCnt = 2;
                      INSERT INTO @NewAll (id, visCnt) (SELECT * FROM @Temp);
                      DELETE FROM @Temp;
                  END;
             SET @NewCnt = (SELECT COUNT(*) FROM @NewAll);
         END;
    SET @connectedCnt = (
         SELECT COUNT(DISTINCT v.id)
         FROM @Visited v
    );
    IF (@connectedCnt > 4 AND @connectedCnt <= 10)
    BEGIN
         SELECT n.paperID, n.paperTitle
         FROM Nodes n, @Visited v
         WHERE n.paperID = v.id
    END;
    DELETE t FROM @tempVisited t JOIN @Visited v ON t.id = v.id;
    DELETE FROM @Visited
    SET @tempVisitCnt = (
         SELECT DISTINCT Count(*)
         FROM @tempVisited
    );
END;
```

END;

[Result]

Table 1

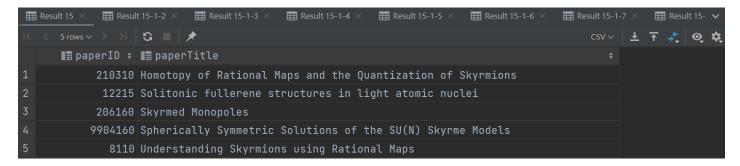


Table 2

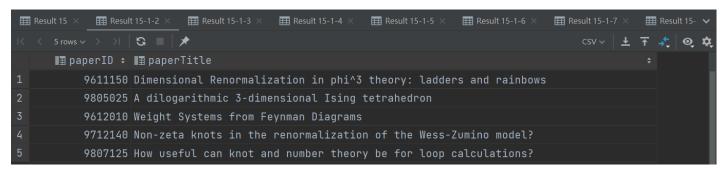


Table 3

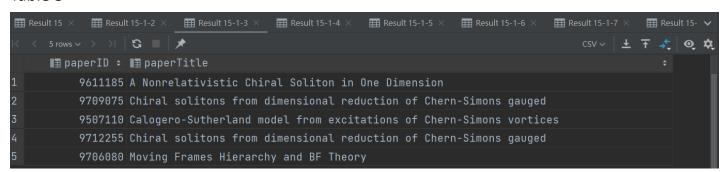


Table 4

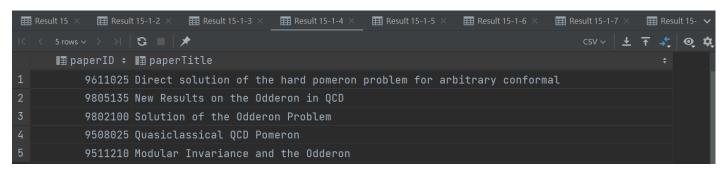


Table 5

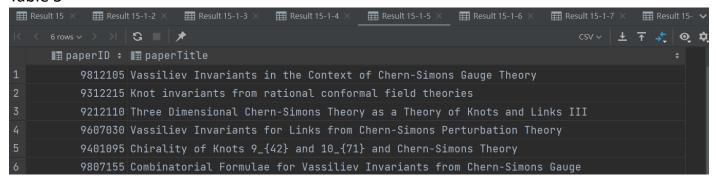


Table 6

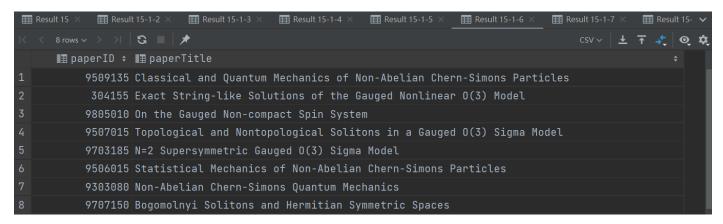


Table 7

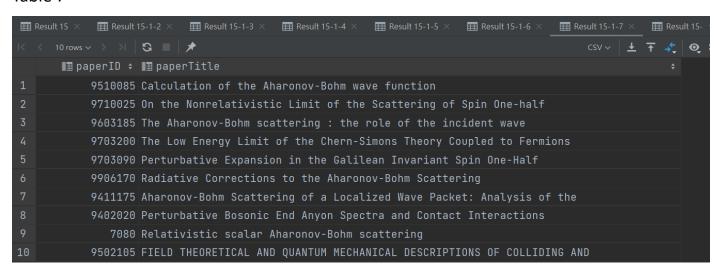
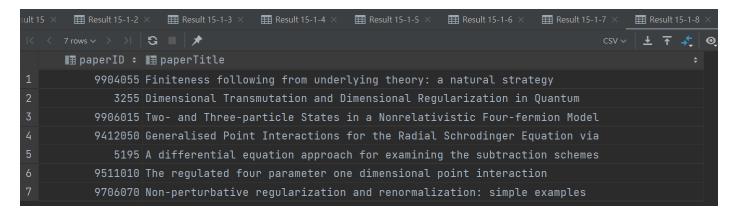


Table 8



1-2 PageRank

[Code]

```
-- 1.2 PageRank
BEGIN
    DECLARE @PageRank TABLE (
         id INT,
         rank FLOAT
    );
    DECLARE @paperCited TABLE (
         id INT
    );
    DECLARE @Visited TABLE (
         id INT
    );
    DECLARE @Sink TABLE (
         id INT
    );
    -- Set up variables
    DECLARE @dampingFactor FLOAT = 0.85;
    DECLARE @nodeCnt FLOAT = (SELECT Count(*) FROM Nodes);
    DECLARE @stayProb FLOAT = ((1 - @dampingFactor) / @nodeCnt);
    DECLARE @sigma FLOAT = 1;
    DECLARE @visitedCnt INT;
    INSERT INTO @sink (id)
    SELECT DISTINCT n.paperID
    FROM Nodes n
    WHERE NOT EXISTS (
         SELECT n1.paperID
         FROM Nodes n1, Edges e
         WHERE n.paperID = n1.paperID AND n1.paperID = e.paperID
    )
    -- Initialize all ranks
    INSERT INTO @PageRank (id) (SELECT n.paperID FROM Nodes n);
    UPDATE @PageRank SET rank = (1 / @nodeCnt);
    WHILE (@sigma > 0.01)
         BEGIN
             INSERT INTO @Visited (id) (SELECT n.paperid FROM Nodes n);
```

```
SET @visitedCnt = (SELECT COUNT(*) FROM @Visited)
SET @sigma = 0;
WHILE (@visitedCnt > 0)
    BEGIN
        -- Randomly pick a PaperID to calculate its PageRank
         DECLARE @curPaperID INT = (
             SELECT TOP(1) v.id
             FROM @Visited v
             ORDER BY NEWID()
        );
        INSERT INTO @paperCited (id)
        SELECT DISTINCT e.paperID
        FROM Edges e
        WHERE e.citedPaperID = @curPaperID
        -- @prevRank = All previous
        DECLARE @prevRank FLOAT = (
             SELECT p.rank
             FROM @PageRank p
             WHERE p.id = @curPaperID);
         DECLARE @updateVal FLOAT = 0;
        DECLARE @paperCitedCnt INT = (SELECT COUNT(*) FROM @paperCited);
         DECLARE @curSinkVal FLOAT = ((
             SELECT SUM(p.rank)
             FROM @Sink s, @PageRank p
             WHERE s.id = p.id) / @nodeCnt) * @dampingFactor;
        WHILE (@paperCitedCnt > 0)
             BEGIN
                 DECLARE @paperID INT = (
                      SELECT TOP (1) p.id
                      FROM @paperCited p
                 );
                 DECLARE @paperIDCitedCnt INT = (
                      SELECT COUNT(e.citedPaperID)
                      FROM Edges e
                      WHERE e.paperID = @paperID
                 );
                 SET @updateVal += (
                      SELECT p.rank
```

```
FROM @PageRank p
                          WHERE p.id = @paperID
                      ) / @paperIDCitedCnt;
                      DELETE FROM @paperCited WHERE id = @paperID;
                      SET @paperCitedCnt -= 1;
                 END;
             SET @updateVal += @curSinkVal
             DECLARE @curRank FLOAT = @stayProb + @dampingFactor * @updateVal;
             SET @sigma += ABS(@curRank - @prevRank);
             UPDATE @PageRank SET rank = @curRank WHERE id = @curPaperID;
             DELETE FROM @Visited WHERE id = @curPaperID;
             SET @visitedCnt -= 1
        END;
    PRINT @sigma;
END;
SELECT TOP(10) p.id, n.paperTitle, p.rank
FROM @PageRank p, Nodes n
WHERE n.paperID = p.id
ORDER BY p.rank DESC
```

[Result]

END;

```
Output
          Result 1
    10 rows ∨ > >| S ■ *
    III id : III paperTitle
                                                                                                     III rank ÷
    9504090 Massless Black Holes and Conifolds in String Theory
                                                                                         0.012402918223824456
    9510135 Bound States Of Strings And p-Branes
                                                                                         0.012096013790628381
    9711200 The Large N Limit of Superconformal Field Theories and Supergravity
                                                                                         0.011524424959577019
    9802150 Anti De Sitter Space And Holography
                                                                                         0.008156857144475966
    208020 Open strings and their symmetry groups
                                                                                         0.007296756063972605
    9602065 D--branes and Spinning Black Holes
                                                                                         0.006377747349660877
    9305185 Duality Symmetries of 4D Heterotic Strings
                                                                                         0.006355169135801032
    9611050 TASI Lectures on D-Branes
                                                                                         0.005999771738579745
    9501030 Strong/Weak Coupling Duality from the Dual String
                                                                                         0.004839057427360183
    9602135 Entropy and Temperature of Black 3-Branes
                                                                                         0.004571579770257256
```