

Aufgabe 1: V-Optimale Histogramme**(1 P.)**

- a) Gegeben folgende Tabelle mit Werten und ihren Häufigkeiten. Berechnen Sie das V-Optimale Histogramm für $B = 2$ Zellen unter Anwendung des in der Vorlesung vorgestellten Algorithmus. Geben Sie dabei verwendete Hilfsstrukturen wie P und PP an, und die DP-Tabelle $SSE^*(i, k)$, insbesondere für Schritte, in denen eine günstigere Möglichkeit gefunden wurde. Geben Sie am Ende die optimalen Histogrammgrenzen an.

Wert	Häufigkeit
1	4
2	2
3	8
4	9
5	3

- b) Zeigen Sie formal folgende Gleichheit:

$$SSE([i, j]) = \sum_{i \leq k \leq j} (F[k]^2) - (j - i + 1) * AVG([i, j])^2$$

- c) Begründen oder widerlegen Sie folgende Aussage:

$$SSE([a, b]) \geq SSE([a, c]) + SSE([c + 1, b])$$

dabei sei $a \leq c < b$.

Aufgabe 2: Wavelets, Sketches**(1 P.)**

- a) Berechnen Sie für folgende kumulative Verteilung über den Zahlen 0 bis 7 die Wavelet-Transformierte und berechnen Sie den Mean-Squared-Error (MSE) in der Approximation durch die Wavelet-Transformierte, die durch Weglassen von Koeffizienten ≤ 1 entsteht.

[6, 7, 8, 9, 13, 14, 15, 30]

- b) Implementieren Sie den **KMV-Sketch** in einer Programmiersprache Ihrer Wahl. Achten Sie darauf, dass die Platzkomplexität in $O(k)$ liegt. Sie dürfen auch das auf der Vorlesungsseite zur Verfügung gestellte Java-Template benutzen. In diesem Template sind Hinweise zur Benutzung.

Führen Sie den Sketch mit $k = 10, 100$ und 1000 für eine Abschätzung der distincten 1.orderkey-Tupel in der TPC-H-Tabelle `lineitem` aus und interpretieren Sie die Ergebnisse.

Aufgabe 3: Statistiken in PostgreSQL

(1 P.)

Das Datenbankmanagementsystem PostgreSQL verwaltet für jede Spalte in jeder Tabelle statistische Daten, die zur Anfrageoptimierung verwendet werden können. Betrachten Sie folgenden Auszug der Statistiken zur Tabelle *lineitem* aus dem TPC-H-Datensatz:

```
SELECT * FROM pg_stats WHERE tablename = 'lineitem'
```

attname	null_frac	n_distinct	most_common_vals	most_common_freqs
l_linenumber	0	7	{1,2,3,4,5,6,7}	{0.25,0.21,0.17,0.14,0.10,0.07,0.03}
l_orderkey	0	419664	{1703939,3644579,507137,703172,712326,770882,971014}	{0.000133333,0.000133333,0.0001,0.0001,0.0001,0.0001}
l_extendedprice	0	-0.11964	{13747.3,25165.2,26677.8,32274.0,33265.3,35938.8,50370.3 }	{0.0001,0.0001,0.0001,0.0001,0.0001,0.0001,0.0001 }

- Was bedeuten die einzelnen Spalten von *pg_stats*?
- Für welche der folgenden Anfragen können diese Statistiken nützlich sein?
 - SELECT * FROM lineitem
 - SELECT * FROM lineitem WHERE l_orderkey IS NOT NULL
 - SELECT * FROM lineitem WHERE l_orderkey = 406631
 - SELECT * FROM lineitem WHERE l_extendedprice = 32274.0
 - SELECT * FROM lineitem WHERE l_extendedprice <= 9500.0
- Schätzen Sie so gut es geht die Anzahl der Ergebnistupel für die Anfragen aus (b) nur anhand der Einträge in *pg_stats*.

Im Folgenden sehen Sie eine weitere Spalte von *pg_stats*, in der ein Histogramm über die Spaltenwerte definiert wird. Achtung: NULL-Werte sowie die Werte aus *most_common_vals* fließen nicht in die Histogrammberechnung ein!¹

attname	histogram_bounds
l_extendedprice	{906.00,1528.60,2062.06,3035.16,3639.44,4444.04,5265.52,5920.60,6751.85,7518.63,8252.80,9032.32,9777.68,10540.68,11267.46,11921.76,12652.00,13413.40,14078.28 (...) }

- Bei welchen Anfragen aus b) sind die Informationen aus der Spalte *histogram_bounds* hilfreich? Schätzen Sie erneut die Anzahl der Ergebnistupel.

Aufgabe 4: Joinordering

(1 P.)

Gegeben folgende Query über der TPC-H Datenbank:

```
SELECT *
FROM part p, partsupp ps, lineitem l
WHERE p.p_partkey = ps.ps_partkey
AND ps.ps_partkey = l.l_partkey AND ps.ps_suppkey = l.l_suppkey
```

- Zeichnen Sie den Anfragegraphen.
- Finden Sie heraus, wie groß die Join-Selektivitäten in dem in der Vorlesung vorgestellten TPC-H Datensatz sind.
- Berechnen Sie für alle möglichen Join-Bäume ohne Kreuzprodukte die Kosten C_{out} .

¹histogram_bounds: "A list of values that divide the column's values into groups of approximately equal population. The values in most_common_vals, if present, are omitted from this histogram calculation." – <https://www.postgresql.org/docs/10/static/view-pg-stats.html>