Generating Permutations
Generating Permutations

Query Optimization: Exercise

Session 8

October 22, 2018

Keep current prefix and the rest of relations

Extend the prefix only if exchanging the last two relations does not result in a cheaper sequence
Transformative Approaches
Explore the search space by directly applying equivalences to the initial join tree [?].
Random Trees with Cross Products
Generate a tree, then generate a permutation: \( C(n - 1) \) trees, \( n! \) permutations

- Pick a random number \( b \in [0, C(n - 1)] \), \textit{unrank} \( b \)
- Pick a random number \( p \in [0, n!] \), \textit{unrank} \( p \)
- Attach the permutation to the leaves of the tree
Unrank($n, r$)

**Input:** the number $n$ of elements to be permuted
and the rank $r$ of the permutation to be constructed

**Output:** a permutation $\pi$

for each $0 \leq i < n$

\[ \pi[i] = i \]

for each $n \geq i > 0$ descending

\{ 
swap($\pi[i - 1], \pi[r \mod i]$)
\[ r = \left\lfloor \frac{r}{i} \right\rfloor \]
\}

return $\pi$;
- every tree is a word in \{ (, ) \}
- map such words to the grid, every step up is (, down )
- the number of different paths \( q \) can be computed (see lectures)
- Procedure: start in (0,0), walk up as long as rank is smaller than \( q \). When it is bigger, step down, \( rank = rank - q \)
Next Homework

- unrank permutation/tree
- implement ExhaustiveTransformation2
- Slides and exercises: db.in.tum.de/teaching/ws1718/queryopt
- Send any questions, comments, solutions to exercises etc. to radke@in.tum.de
- Exercise due: 9 AM, December 18