A Functional Approach to External Graph Algorithms [ABW98]
James Abello, Adam L. Buchsbaum, Jeffery R. Westbrook

Presenter: Amartya Shankha Biswas

February 25, 2019
Number of Block Transfers to/from External Memory

\[ \text{scan}(N) \equiv O\left(\left\lceil \frac{N}{B} \right\rceil\right) \]
\[ \text{sort}(N) \equiv O\left(\text{scan}(N) \log_b \frac{N}{B}\right) \]
Functional I/O model for Graph Algorithms

- Sequence of functions applied to input
  - No side effects
  - Easy to enforce write-once discipline
- Simple data-structures are difficult to implement
- Batch updates and node copying add to I/O and space complexity

- Graphs in the external model
  - Represented as list of edges
  - No adjacency list
- Semi-external Model
  - Vertices fit in internal memory
  - $|V| < M < |E|$
Building Blocks

- Selection
- Relabeling
- Contraction
Select($I, k$) using median-of-medians in $\mathcal{O}(\text{scan}(I))$

Recursively, select medians of medians of subarrays of size $M$.

In expectation, $T(N) = T\left(\frac{3}{4}N\right) + \mathcal{O}\left(\frac{N}{B}\right) \Rightarrow T(N) = \mathcal{O}\left(\frac{N}{B}\right)$
Relabel \((I, F)\) in \(O(sort(I) + sort(F))\)

- Sort edges in \(I\) by first endpoint and edges in \(F\) by source
- Iterate through sorted lists in tandem, relabeling first endpoints of \(I\)
- Repeat for second endpoint by sorting \(I\) again
Contract \( (I, \{C_1, C_2, \cdots \}) \) in \( \mathcal{O} \left( sort(I) + sort(\sum |C_i|) \right) \)

Reduce to Relabeling

- Replace each \( C_i \) with a star \( S_i \)
- Concatenate \( \langle S_1, S_2, \cdots \rangle \) to obtain \( F \)
- Contract \( (I, \{C_1, C_2, \cdots \}) = \text{Relabel}(I, F) \)
Graph Partitioning: Divide-and-Conquer

- How to partition edges?
- Which sub-problem to solve?
- How to recombine?
Connected Components

- Sample half the edges of the graph as $E_1$
- Recursively compute $C_1 = \text{Connected-Components}(G_1 = (V, E_1))$
- Use contraction to obtain $G_2 = \text{Contract}(G, C_1)$
- Recursively compute $C_2 = \text{Connected-Components}(G_2)$
- Return Connected-Components($G$) = $C_2 \cup \text{Relabel}(C_2, C_1)$

\[
T(N) = 2T\left(\frac{N}{2}\right) + \mathcal{O}(\text{sort}(|E|))
\]

- Repeat until problem size $\leq M$
- $\log_2 \frac{|E|}{M}$ iterations
- Total I/O complexity $T(|E|) = \mathcal{O} \left(\text{sort}(|E|) \cdot \log_2 \frac{|E|}{M}\right)$
Find the median edge weight $m$ by running Select$(E, |E|/2)$

Compute $E_1 \subset E$ as the set of edges with weight $\leq m$

Recursively compute $T_1 = \text{MST}(G_1 = (V, E_1))$

Compute the connected components of the MST obtained using half the edges: $C_1 = \text{Connected-Components}(T_1)$

Use contraction to obtain $G_2 = \text{Contract}(G, C_1)$

Recursively compute $T_2 = \text{MST}(G_2)$

Return $T = T_1 \cup \text{Inverse-Relabel}(T_2, C_1)$
Maximal Matching

- Sample half the edges of the graph as $E_1$
- Recurse to find $M_1 = \text{Maximal-Matching}(G_1 = (V, E_1))$
- Find set of vertices covered by the matching $V_1 = V(M_1)$
- Let $E_2 = E \setminus (V_1 \times V_1)$ and $G_2 = (V, E_2)$
- Return $M = M_1 \cup \text{Maximal-Matching}(G_2)$
## Minimum Spanning Tree
- Maintain union-find data structure in memory
- Run Kruskal’s algorithm

## Connected Components
- How to re-arrange edges contiguously by component?
- I/O complexity dominated by sorting: 
  \[ O \left( \text{scan}(|E|) \cdot \log_b \frac{|E|}{B} \right) \]
- What if there are few connected components?
- Desired runtime: 
  \[ O \left( \text{scan}(|E|) \cdot \log_b |C| \right) \]
- \(|C|\) is # of connected components
Grouping \( N \) Elements with keys in range \([1 \ldots G]\)

Use \( b \) blocks in internal memory

- Each block stores elements from a disjoint range of length \( G/b \)
- Blocks are emptied to external memory when full

Recurse on each range (size \( G/b \)) from the last step

Sub-divide into three sub-ranges of size \( G/b^2 \) and so on . . .

Done after \( \mathcal{O}(\log_b G) \) iterations

Total I/O complexity = \( \mathcal{O}(\text{scan}(N) \cdot \log_b G) \)
Grouping with $b = 3$ and $G = 27$

Partially filled blocks?

Concatenate to ensure that there is at most one.
Discussion

- Other graph problems:
  - Shortest paths
  - Random walk

- Assume properties of the ordering of edges
James Abello, Adam L Buchsbaum, and Jeffery R Westbrook.
A functional approach to external graph algorithms.