1. Recall from Yongyi’s lecture: Single-Source Shortest Paths (SSSP) - find shortest paths to all vertices from a source vertex $s$.
   (a) Unweighted graphs: parallel BFS to depth $k$ in $O\left(\frac{m}{p} + k \log^* p\right)$ w.h.p.
   (b) Integer-weighted graphs: treat edge $uv$ of length $l$ as $l$ unweighted edges, then use parallel unweighted BFS.
      i. How to keep work dependent on $m$ instead of on sum of all edge lengths?
      ii. Don’t explore each edge individually; mark $v$ as $l$ levels deeper than $u$.

2. Today: FullSearch - $k$-limited, $\frac{1}{2}$-approximation SSSP for weighted graphs
   (a) $k$-limited: $s$-$t$ path has length at most the length of any $s$-$t$ path of size $k$.
   (b) $k$-limited, $\frac{1}{2}$-approximate: $s$-$t$ path has length at most $1\frac{1}{2}$ times the length of any $s$-$t$ path of size $k$.
   (c) Work: $O(m \log D)$, $D =$ sum of all edge lengths.
   (d) Depth: $O\left(\frac{m \log D}{p} + k\right)$, $p =$ number of processors.

3. FullSearch’s subroutine Search
   (a) FullSearch, but with extra parameter $d$ such that every path returned has length at least $d$ and at most $2d$.
   (b) Work: $O(m)$; Depth: $O\left(\frac{m}{p} + k\right)$.
   (c) Approximate $G$ to integer-weighted $\hat{G}$ by rounding edge lengths up to the nearest $\alpha$.
      \[
      \hat{l}(e) = \begin{cases} 
      \alpha \left\lceil \frac{l(e)}{\alpha} \right\rceil & \text{if } l(e) > 0 \\
      \alpha & \text{else } l(e) = 0
      \end{cases}
      \]
   (d) Solve SSSP for $\hat{G}$ using integer-weighted BFS.
   (e) Intuitively, $\alpha$ is a guess at how short an edge in the shortest distance path is.
      i. $d$ is a guess at how short the total path is.
      ii. $\alpha \sim \frac{d}{k}$, the average edge length of the shortest distance path of size $\leq k$.
(f) Correctness lemma: If $P$ is an $s$-$t$ path of size $k$ and length $x$, $d \leq x \leq 2d$, then the estimated distance to $t$ is at most $1\frac{1}{2}$ times the length of $P$.

Proof. We prove that the estimated path length is a $\frac{1}{2}$-approximation, if $\alpha = \frac{d}{2k}$:

\[
\hat{l}(e) \leq l(e) + \alpha \\
\hat{l}(P) \leq l(P) + k\alpha \\
\hat{l}(P) \leq \left(\frac{3}{2}\right) l(P)
\]

We also prove that a BFS to some depth will find the estimated path in $\hat{G}$:

\[
\hat{l}(P) \leq 3d = 6k \left(\frac{d}{2k}\right) = 6k(\alpha)
\]

Magic! A depth of $6k$ works.

(g) Work: $O(m)$ because of integer-weighted BFS.

(h) Depth: $O\left(\frac{m}{p} + k\right)$ because there are $p$ processors and BFS depth is $6k$.

4. FullSearch, continued

(a) Call Search with $d = 1, 2, 4, \ldots D$, $D$ is the sum of all edge lengths.
(b) Work: $O(m \log D)$ because there are $\log D$ calls to Search.
(c) Depth: $O\left(\frac{m \log D}{p} + k\right)$ because there are $p$ processors and BFS depth is $6k$.

5. Coming soon: Reduction from exact SSSP to approximate SSSP and algorithm for approximate SSSP using FullSearch.