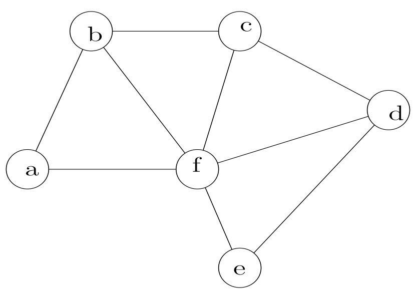
CompSci 711—(2021 S2)

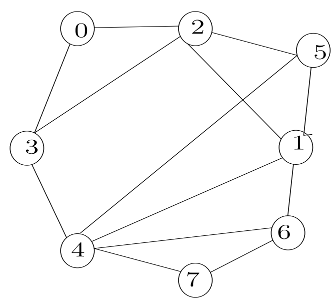
Assignment 1 (Parallel Computing)

Due: 7 Aug 2021

This assignment covers the theory and design of parallel algorithms. It is worth 15% of your total grade for the course. Please upload your answers in a PDF file to Canvas. The use of a professional typesetting software such as LaTeX is recommended.

1. (Parallel Architecture) For the two hypothetical parallel architectures given below answer the following:



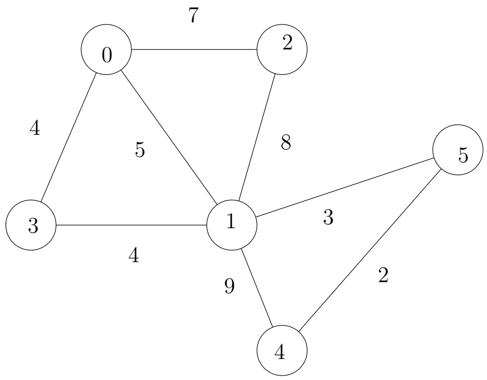


   (a) What is the diameter, (arc) connectivity and bisection width of each? **[marks 5]**

   (b) Find an efficient embedding of the graph on the left into the graph on the right. What is the dialation and congestion of your embedding (topology mapping)? **[marks 5]**

2. (Bitonic Sorting) Consider a bitonic sequence $s = [a_0, a_1, \ldots, a_{n-1}]$, where n is a power of 2. Prove that the sequences s_1 and s_2 obtained from s by performing the bitonic split operation described in the course slides (that is, $s_1 = [\min(a_0, a_{n/2}), \ldots, \min(a_{n/2-1}, a_{n-1})]$ and $s_2 = [\max(a_0, a_{n/2}), \ldots, \max(a_{n/2-1}, a_{n-1})]$) satisfy the properties (1) s_1 and s_2 are bitonic sequences, and (2) the elements of s_1 are all smaller than the elements of s_2. **[marks 10]**

3. (Parallel Graph Algorithms) For the following edge-weighted graph, illustrate (via traces and computional time) the difference between the source-partitioned and source-parallel Dijkstra's algorithm for computing all-pairs shortest paths (e.g. distance matrix). Let the number of processors $p = 24$.



[**marks 10**]

4. (P Systems) Consider a tree-based P system $\Pi = (O, K, \delta)$ with cell states:

   (a) O is a finite non-empty alphabet of *objects* (no cell-IDs);

   (b) $K = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ is a finite set of *cells*;

   (c) δ is rooted tree at cell σ_1.

   Each cell, $\sigma_i \in K$, has the initial configuration $\sigma_i = (Q_i, s_{i0}, w_{i0}, R_i)$, and the current configuration $\sigma_i = (Q_i, s_i, w_i, R_i)$, where:

   - Q_i is a finite set of *states*;
   - $s_{i0} \in Q_i$ is the *initial state*; $s_i \in Q_i$ is the *current state*;
   - $w_{i0} \in O^*$ is the *initial content*; $w_i \in O^*$ is the *current content*;
   - R_i is a finite *ordered* set of multiset rewriting *rules* of the form: $s\ x \rightarrow_\alpha s'\ x'\ (u)_\beta$, where $s, s' \in Q$, $x, x', u \in O^*$, $\alpha \in \{\mathtt{min}, \mathtt{max}\}$, and $\beta \in \{\uparrow, \downarrow, \updownarrow\}$. For convenience, we allow a rule to contain zero or more instances of $(u)_\beta$.

   The rules are applied in the *weak priority* order, i.e. (1) higher priority applicable rules are applied before lower priority applicable rules, and (2) a lower priority applicable rule is applied only if it indicates the same target state as the previously applied rules.

   The rewriting operator $\alpha = \mathtt{max}$ indicates that an applicable rule of R_i is applied as many times as possible (otherwise just once, if $\alpha = \mathtt{min}$). If the right-hand side of a rule contains $(u)_\beta$, $\beta \in \{\uparrow, \downarrow, \updownarrow\}$, then for each application of this rule, a copy of multiset u is replicated and sent to each cell $\sigma_j \in \delta^{-1}(i)$ if $\beta = \uparrow$, $\sigma_j \in \delta(i)$ if $\beta = \downarrow$ and $\sigma_j \in \delta(i) \cup \delta^{-1}(i)$ if $\beta = \updownarrow$.

   Develop a set of rules so that the system terminates with cell σ_1 containing the height of its rooted tree, encoded as the multiplicity of object $h \in O$. Hint: Do a broadcast then convergecast, then divide by two the total time taken. [**marks 10**]

5. (Randomized Parallel Algorithms) Consider Luby's Maxim<u>al</u> Independent Set Algorithm as a template. Explain how it can be adapted (if possible) to the following two problems which take input a graph $G = (V, E)$:

   (a) Minimal Vertex Cover: find $V' \subseteq V$ such that (1) $G - V'$ contains no edges and (2) for all $v'' \subset V'$, $G - V''$ contains at least one edge. [**marks 5**]

   (b) Minimal Feedback Vertex Set: find $V' \subseteq V$ such that (1) $G - V'$ contains no cycles and (2) for all $V'' \subset V'$, $G - V''$ contains at least one cycle. [**marks 5**]