2/19/09
CUDA memory I

Review:
- list compaction app
  - example: image adaptive sampling
- scan applications
- first order recurrence problems
  - discussed up-sweep \( \rightarrow \) reduction with a general
    - down-sweep would be the same
      - associative operator

- high order recurrence:

\[
\begin{align*}
  x_i &= \sum_{j=0}^{i-1} b_j \\
  &\quad + \left( x_{i-1} \times a_{i,1} \right) + \left( x_{i-2} \times a_{i,2} \right) + \ldots + (x_{i-m} \times a_{i,m}) \\
  &+ b_i \quad \text{recurr}
\end{align*}
\]

Let's define a vector form of \( x_i \) as \( \vec{x}_i \)

\[
\vec{x}_i = [x_i, x_{i-1}, x_{i-2}, \ldots, x_{i-m+1}]
\]

Then we can cast this problem as a first order recurrence as:

\[
\vec{x}_i = [x_i, x_{i-1}, x_{i-2}, \ldots, x_{i-m+1}] \begin{bmatrix}
  a_{i,0} \\
  a_{i,1} \\
  \vdots \\
  a_{i,m-1} \\
  a_{i,m} \end{bmatrix} + [b_i, 0, 0, 0, \ldots, 0]
\]

\[
= \vec{x}_{i-1} \times A + \vec{b}_i
\]

\[\text{vector-matrix mult.} \]
a quick look at Fibonacci sequence:

\[ x_i = \begin{cases} \begin{bmatrix} 1 \\ 0 \end{bmatrix}, & i = 0, 1 \\ x_{i+1} + x_{i-2}, & i \geq 2 \end{cases} \]

\[ \overrightarrow{x_i} = [x_i, x_{i-1}] \]

\[ A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \]

so:

\[ \overrightarrow{x_i} = [x_i, x_{i-2}] \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = x_{i-1} \times A \]

\[ \Rightarrow \text{ no constant term.} \]

so in fact:

\[ \overrightarrow{x_i} = A^i, \quad i \geq 2 \]

so what you do is:

matrix multiplication:

\[ \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \cdots \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \]

\# of operations?
- Segmented scan:
  
  resets where the seg-tag is 1.
  Cast as a first order recurrence problem:

  \[ x_i = \begin{cases} b_0 & i = 0 \\ (1-f) \times x_{i-1} + b_1 & i > 1 \end{cases} \]

Example: quicksort using segmented scan.

Keep each subset in its own segment.
Pick pivot values and split the keys independently within each segment.