The "accept-reject" method

(AKA rejection sampling)
Given 2-dimensional domain

\[
\begin{array}{c}
\text{Goal: output random blue block} \\
\text{Naive rejection sampling algorithm:} \\
\text{Repeat Forever:} \\
\text{Pick random } x \in [x_{\text{max}}] \\
\text{Pick random } y \in [y_{\text{max}}] \\
\text{If } (x, y) \text{ is a blue block, output } (x, y, y) \\
\text{halt}
\end{array}
\]

Analysis: output \((x, y, y)\) with prob \(\frac{1}{\text{\# blue blocks}}\)

Expected runtime per output: \(\frac{1}{\text{fraction covered by blue blocks}} = \frac{x_{\text{max}} \cdot y_{\text{max}}}{\text{\# blue blocks}}\)
Better algorithm? Assume we know $b_x$ and there are blue blocks at every $x$. 

Repeat forever:
- Pick random $x \in [X_{max}]$.
- Pick random $y \in [b_x]$. 
- Toss coin with prob $\frac{b_x}{b_x}$. 
- If heads, output $(x,y)$ and halt.

Analysis:
- Pick $(x,y)$ with prob $\frac{1}{X_{max} \cdot b_x}$.
- Output $(x,y)$ with prob $\frac{1}{X_{max}} \cdot \frac{b_x}{b_x} \cdot \frac{b_x}{b_y} = \frac{1}{X_{max} \cdot b_y}$. 

Expected runtime per output: $\frac{X_{max} \cdot b_y}{\# \text{blue blocks}}$. 

When is this better? E.g., if average $b_y \geq \frac{1}{2} \cdot \max b_y$. Then this is 2.
Example in class:

After bucketing nodes by degree,

within a bucket all nodes had degree

\[
(1+\beta)^i \leq \text{degree} \leq (1+\beta)^{i+1}
\]

So \(\max\) degree \(\leq (1+\beta)^{i+1}\)

ave degree \(\geq (1+\beta)^i\)