Shortest path computation using matrix representation (Adjacency matrix)

\[ A = \begin{pmatrix}
  a & b & c & d \\
  b & 10 & 0 & 0 \\
  c & 9 & 8 & 0 \\
  d & 8 & 0 & 0 \\
\end{pmatrix} \]

\[ \infty + \infty = \infty \]

\[ A \odot A = B : B_{ij} = \min_k \left\{ A_{ik} + A_{kj} \right\} \]

\[
\begin{pmatrix}
  0 & \infty & \infty & \\infty \\
  10 & 0 & \infty & 3 \\
  9 & \infty & 0 & \infty \\
  6 & \infty & \infty & 0 \\
\end{pmatrix}
\]

\[ A = \begin{pmatrix}
  a & 0 & 0 & 0 \\
  0 & 10 & 0 & 0 \\
  9 & 0 & 0 & 0 \\
  6 & 0 & 0 & 0 \\
\end{pmatrix} \]

Claim: \[ A = A \oplus A \oplus \ldots \oplus A \] gives us \[ i \] shortest paths with at most \[ i \] edges
$A^{n-1}$ gives us APSP

Dijkstra's algorithm uses Heap

- Min of all labels: Extract min
- Change (decrease) labels
  - Using Heaps: $O(\log n)$ for both operations

Heap: min, Extract min, insert, delete, (priority queues) decrease $\rightarrow$ delete + insert

Given 2 Heaps $H_1$ and $H_2$ create a new heap $H_1 \cup H_2$

Given 2 dictionaries, how do we combine them?

Simplest scheme will be to insert element by element from the 'smaller' to the 'larger':

$|H_1| = \sqrt{n}$ $\rightarrow$ $H_2$ : $n - \sqrt{n}$

$H_1 \rightarrow H_2$ $\sqrt{n} \cdot \log (n - \sqrt{n}) < \sqrt{n} \log n$
$t_2 \rightarrow t_1, \quad n \cdot \log_2 (\sqrt{n})$
\[
\leq \frac{n \log n}{2} - \frac{\sqrt{n} \log n}{2}
\]

Notice that constructing a heap from scratch takes $O(n)$ time.

Can we combine heaps in $O(n \log n)$?

```
H1
```

```
H2
```

Take any element from the last level (delete)
create a new root node.
Then call `Heapify`:
```
Total time is $O(n \log n)$
```

```
linear structure
```

Heap property: (min heap)
- the value at a node
  is no larger than children
Can we improve the performance of a heap using a $k$-ary structure? Normal heap: binary

Does it lead to any improvements? ($\log kn$)

Consider a family of trees $B_i$ of the following kind:

$B_0$

$B_{i+1}$ (tree of $i+1$ nodes) Given two $B_i$ trees

- make the root $T$ one copy of $B_i$, the left child of the other copy $B_1$
- $B_2$

$(\frac{3}{1}): 2 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}$
How many nodes in $B_i$? $2^i$

The family of trees is called Binomial Trees

**Properties**

1. $B_i$ has $2^i$ nodes
2. The root of $B_i$ has $i$ children
3. The depth of $B_i$ is $i$
4. $B_n$ has $\sum_{j=0}^{\lfloor \log_2 n \rfloor} \binom{n-1}{j} + \binom{n-1}{j-1} = \binom{n}{j}$ nodes in depth $j$

$B_{n-1}$ $B_{n-1}$

$\binom{n-1}{j} + \binom{n-1}{j-1} = \binom{n}{j}$

level $j$ has one copy, level $j-1$ has the other
Binomial Heap

is a collection of ordered Binomial Trees such that there are no more than one $B_j$ for each $j$ and the roots are linked in a list.

$B_0 \quad \times \quad B_2 \quad B_4$

- where is the min? One of the root nodes
- Cost: no of trees in the heap
- Extract min; Do extract min on the tree corresponding to min

For a heap of $n$ values, how many Binomial trees are needed?

$n$ is not a power of 2

Binary representation of $n = 8 + 2 + 1 \iff B_3 B_2 B_0$