Prim: grow a connected set $A$ into a MST
- want to easily find light edge across cut $(v \in A, \text{else})$
- maintain, for each $v$, distance to vertex in $A$
  total weight 17
d(v): weight of min edge from v to v in G

Pseudocode to find min d(v)

Init
Pick starting vertex s
\[ d[v] = \infty \text{ for each } v \in V \]
\[ d[s] = 0 \]
\[ \pi[v] = NIL \text{ for each } v \in V \]

Put all \( v \in V \) in P.Q. using key \( d[v] \) (the edge \((v, \pi[v])\) has weight \( x \))

While (Q \( \neq \emptyset \))

\[ v = Q\text{.extract-min()} \]

Add \((v, \pi[v])\) to A

For each \( u \in \text{Adj}[v] \)

\[ d[u] > w(v, u) \text{ and } u \in Q \]
\[ d[u] = w[v, u] \]
\[ \pi[u] = v \]
\[ Q\text{.decrease-key}(u, d[u]) \]
<table>
<thead>
<tr>
<th>Operation</th>
<th># times</th>
<th>Binary heap</th>
<th>Unsorted array</th>
<th>Fib-heap</th>
</tr>
</thead>
<tbody>
<tr>
<td>build-queue</td>
<td>1</td>
<td>$O(V)$</td>
<td>$O(V)$</td>
<td>$O(V)$</td>
</tr>
<tr>
<td>extract-min</td>
<td>$V$</td>
<td>$O(\log V)$</td>
<td>$O(V)$</td>
<td>$O(\log V)$</td>
</tr>
<tr>
<td>decrease-key</td>
<td>$E$</td>
<td>$O(\log V)$</td>
<td>$O(1)$</td>
<td>$O(1)\text{smooth}$</td>
</tr>
<tr>
<td>total</td>
<td></td>
<td>$O(E \log V)$</td>
<td>$O(V^2)$</td>
<td>$O(E + V \log V)$</td>
</tr>
</tbody>
</table>

Better for sparse graphs, better for dense graphs.

Shortest paths:
1) given $u,v$ what's the shortest path
2) given $v$, shortest path to all else
3) for all pairs, what's shortest path
4) All pairs shortest paths