**2253206 韩明洋**

**初始步骤**

1. **起点选择**
   * 选择城市1作为起点，初始路径为[1]，当前路径长度为0。
2. **计算初始下界**
   * 初始下界估计：从城市1出发访问其他城市的最小路径总长。例如，路径1→4→2→3→1的长度为4 + 10 + 5 + 6 = 25。

**构建状态空间树**

1. **扩展根节点 [1]**
   * 从城市1可以前往城市2、3或4，生成三个子节点：
     + [1→2]，路径长度30。
     + [1→3]，路径长度6。
     + [1→4]，路径长度4。
   * 计算各子节点的下界：
     + [1→2]：下界为30 + 29 = 59（路径2→3→4→1的估计长度为5 + 20 + 4 = 29）。
     + [1→3]：下界为6 + 19 = 25（路径3→2→4→1的估计长度为5 + 10 + 4 = 19）。
     + [1→4]：下界为4 + 21 = 25（路径4→2→3→1的估计长度为10 + 5 + 6 = 21）。
   * **优先扩展下界最小的节点 [1→3] 和 [1→4]**。
2. **扩展节点 [1→3]**
   * 从城市3可以前往城市2或4，生成两个子节点：
     + [1→3→2]，路径长度为6 + 5 = 11。
     + [1→3→4]，路径长度为6 + 20 = 26。
   * 计算下界：
     + [1→3→2]：下界为11 + 14 = 25（路径2→4→1的估计长度为10 + 4 = 14）。
     + [1→3→4]：下界为26 + 9 = 35（路径4→2→1的估计长度为10 + 30 = 40，但路径不完整，实际下界可能更高，此处暂估）。
   * **优先扩展下界最小的节点 [1→3→2]**。
3. **扩展节点 [1→3→2]**
   * 从城市2可以前往城市4，生成子节点：
     + [1→3→2→4]，路径长度为11 + 10 = 21。
   * 计算下界：21 + 4 = 25（路径4→1的长度为4）。
   * **发现完整回路**：路径1→3→2→4→1，总长度为6 + 5 + 10 + 4 = 25。
4. **扩展节点 [1→4]**
   * 从城市4可以前往城市2或3，生成两个子节点：
     + [1→4→2]，路径长度为4 + 10 = 14。
     + [1→4→3]，路径长度为4 + 20 = 24。
   * 计算下界：
     + [1→4→2]：下界为14 + 11 = 25（路径2→3→1的估计长度为5 + 6 = 11）。
     + [1→4→3]：下界为24 + 7 = 31（路径3→2→1的估计长度为5 + 30 = 35，但路径不完整，实际下界可能更高，此处暂估）。
   * **优先扩展下界最小的节点 [1→4→2]**。
5. **扩展节点 [1→4→2]**
   * 从城市2可以前往城市3，生成子节点：
     + [1→4→2→3]，路径长度为14 + 5 = 19。
   * 计算下界：19 + 6 = 25（路径3→1的长度为6）。
   * **发现完整回路**：路径1→4→2→3→1，总长度为4 + 10 + 5 + 6 = 25。

**剪枝**

1. **剪枝操作**
   * 后续节点的下界均≥25，无法产生更优解，故剪枝。

**最终答案**

最短回路的总长度为 **25**，对应路径为：  
1→3→2→4→1 或 1→4→2→3→1。

同济大学计算机科学与技术学院