

Parte 1 Opcional: Búsqueda Manual (Ramificación y Acotación)

Como ejemplo simularemos la búsqueda manual tomando como origen **Arad** y como destino **Bucharest**:

Importante: Cada nodo tiene a su lado su coste acumulado correspondiente.

1. Iteración 1:

- **Nodo actual:** Arad (0)
- **Pop :** Arad(0)
- **Hijos:** Zerind(75), Timisoara(118), Sibiu(140)
- **Lista cerrada:** [(0, Arad)]
- **Lista abierta:** [(75, Zerind), (118, Timisoara), (140, Sibiu)]

2. Iteración 2:

- **Pop:** Zerind (75)
- **Nodo actual:** Zerind(75)
- **Hijos:** Oradea (71) → $(75+71 = 146)$
- **Lista cerrada:** [(0, Arad), (75, Zerind)]
- Lista abierta: [(118, Timisoara), (140, Sibiu), (146, Oradea)]

3. Iteración 3:

- **Pop** Timisoara (118)
- **Nodo actual:** Timisoara (118)
- **Hijos:** Lugoj (111) → $(118+111 = 229)$
- **Lista cerrada:** [(0, Arad), (75, Zerind), (118, Timisoara)]
- **Lista abierta:** [(140, Sibiu), (146, Oradea), (229, Lugoj)]

4. Iteración 4:

- **Pop:** Sibiu (140)
- **Nodo actual:** Sibiu (140)
- **Hijos:** Fagaras (99) → $(140+99=239)$, Rimnicu (80) → $(140+80=220)$
- **Lista cerrada:** [(0, Arad), (75, Zerind), (118, Timisoara), (140, Sibiu)]
- **Lista abierta:** [(146, Oradea), (220, Rimnicu), (229, Lugoj), (239, Fagaras)]

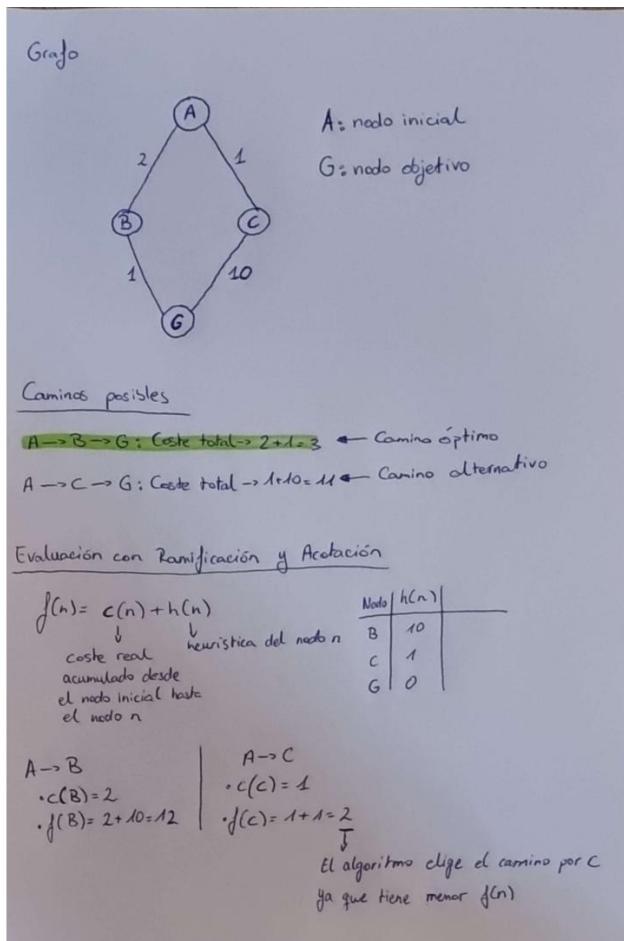
5. **Iteración 5:**

- **Pop:** Oradea (146)
- **Nodo actual:** Oradea (146)
- **Hijos:** Sibiu (151) → (146+151=297)
- **Lista cerrada:** [(0, Arad), (75, Zerind), (118, Timisoara), (140, Sibiu)]
 - El nodo Sibiu ya fue visitado con un coste menor, cualquiera que llegue con coste mayor no se expande, porque nunca mejorará la solución. Por este motivo la nueva ruta que es más cara que la anterior se descarta.
- **Lista abierta:** [(220, Rimnicu), (229, Lugoj), (239, Fagaras), ...]

Parte 2 Opcional: Heurística que sobreestima

Si usamos una heurística que **sobreestima** el coste real ($h(n) > h^*(n)$), el algoritmo A* pierde la propiedad de **admisibilidad**, y por lo tanto puede descartar caminos óptimos creyendo erróneamente que son peores de lo que realmente son.

Ejemplo teórico:



Como vemos en la imagen el algoritmo sigue el camino A->C->G con coste total 11, descartando el camino óptimo A->B->G con coste 3.

Por tanto, sacamos como conclusión que una heurística que sobreestima puede llevar al algoritmo a no encontrar el camino óptimo. Por eso las heurísticas admisibles garantizan optimalidad mientras que las heurísticas que sobreestiman rompen esa garantía,