

Análisis de Optimización de Código (Dijkstra)

Basado en la implementación original en Semana6.py.

1. Eficiencia

- **Uso de visited array:** La implementación actual usa un array `visited` y verifica `if visited[u]: continue`. En Dijkstra con `heapq`, es más común y a veces más eficiente verificar si la distancia extraída es mayor que la distancia ya conocida (`if cost > dist[u]: continue`). Esto evita procesar nodos que fueron añadidos a la cola con una distancia peor antes de encontrar una mejor.
- **Estructura de Grafo:** El uso de `self.adj = {i: [] for i in range(n)}` es correcto, pero si los nodos no son enteros consecutivos 0..n-1, fallará. Usar `defaultdict(list)` o un diccionario normal permite nodos con cualquier identificador (strings, IDs no consecutivos).

2. Claridad

- **Type Hinting:** Faltan tipos específicos en algunas partes.
- **Nombres de Variables:** `u`, `v`, `w` son estándar en teoría de grafos, pero `src`, `cost` podrían ser más descriptivos (`start_node`, `current_dist`).

3. Robustez

- **Validación de Nodos:** No verifica si el nodo `src` existe o está dentro del rango válido antes de acceder a `dist[src]`.
- **Grafos Desconectados:** El código maneja nodos inalcanzables dejándolos como `math.inf`, lo cual es correcto.

4. Extensibilidad

- **Reconstrucción de Camino:** El método devuelve `parent`, pero no incluye una función auxiliar para reconstruir el camino fácilmente desde el destino hasta el origen.
- **Bidireccional:** No soporta grafos dirigidos/no dirigidos explícitamente en la inicialización (está comentado).

Mejoras Propuestas

1. **Optimización de Bucle Principal:** Eliminar visited array y usar la comprobación de distancia ($\text{dist}[u] < \text{cost}$).
2. **Flexibilidad de Nodos:** Permitir nodos que no sean enteros 0..N-1 usando un diccionario para dist y parent.
3. **Helper de Camino:** Añadir método `get_path(target)`.

```
# Ejemplo de Mejora 1: Optimización de Bucle
while pq:
    d, u = heapq.heappop(pq)
    if d > dist[u]: # Lazy deletion
        continue
    # ... resto del código
```