

```

from collections import deque

def bfs(residual, origem, destino, caminho):
    visitado = [False] * len(residual)
    fila = deque([origem])
    visitado[origem] = True

    while fila:
        u = fila.popleft()
        for v, capacidade in enumerate(residual[u]):
            if not visitado[v] and capacidade > 0:
                visitado[v] = True
                caminho[v] = u
                if v == destino:
                    return True
                fila.append(v)
    return False

def edmonds_karp(grafo, origem, destino):
    n = len(grafo)
    residual = [linha[:] for linha in grafo]
    caminho = [-1] * n
    fluxo_maximo = 0

    while bfs(residual, origem, destino, caminho):
        fluxo = float("inf")
        v = destino
        while v != origem:
            u = caminho[v]
            fluxo = min(fluxo, residual[u][v])
            v = caminho[v]

        v = destino
        while v != origem:
            u = caminho[v]
            residual[u][v] -= fluxo
            residual[v][u] += fluxo
            v = caminho[v]

        fluxo_maximo += fluxo

    return fluxo_maximo

grafo = [
    [0, 16, 13, 0, 0, 0],
    [0, 0, 10, 12, 0, 0],
    [0, 4, 0, 0, 14, 0],
    [0, 0, 9, 0, 0, 20],

```

```
[0, 0, 0, 7, 0, 4],  
[0, 0, 0, 0, 0, 0],  
]  
origem = 0  
destino = 5  
  
print("Fluxo máximo:", edmonds_karp(grafo, origem, destino))
```