模版·分支定界法(分配问题)

```java
public class TaskAssignment{
    private static class Node implements Comparable<Node>{
        int[] path;
        int cost;
        int worker;
        int task;

        public Node(int[] path,int cost,int worker,int task){
            this.path=Arrays.copyOf(path,path.length);
            this.cost=cost;
            this.worker=worker;
            this.task=task;
        }

        @Override
        public int compareTo(Node other){
            return Integer.compare(this.cost,other.cost);
        }
    }

    private int[][] costMatrix;
    private int n;

    public TaskAssignment(int[][] costMatrix){
        this.costMatrix=costMatrix;
        this.n=costMatrix.length;
    }

    private int calculateCost(int[] path){
        int totalCost=0;
        for(int i=0;i<path.length;i++){
            if(path[i]!=-1){
                    totalCost+=costMatrix[i][path[i]];
                }
            }
            return totalCost;
        }

        public int findMinCost(){
            PriorityQueue<Node> pq=new PriorityQueue<>();
            int[] initialPath=new int[n];
            Arrays.fill(initialPath,-1);

            Noderoot=new Node(initialPath,0,-1,-1);
            pq.add(root);

            int minCost=Integer.MAX_VALUE;

            while(!pq.isEmpty()){
                Node minNode=pq.poll();

                if(minNode.worker==n-1){
                    minCost=Math.min(minCost,minNode.cost);
                    continue;
                }

                for(int i=0;i<n;i++){
                    if(!isAssigned(minNode.path,i)){
                        int[] newPath=Arrays.copyOf(minNode.path,n);
                        newPath[minNode.worker+1]=i;

                        int newCost=calculateCost(newPath);

                        if(newCost<minCost){
                            pq.add(new Node(newPath,newCost,minNode.worker+1,i));
                        }
                    }
            }
        }

        return minCost;
    }

    private boolean isAssigned(int[] path,int task){
        for(int i:path){
            if(i==task){
                return true;
            }
        }
        return false;
    }

    public static void main(String[] args){
        int[][] costMatrix={
            {9,2,7,8},
            {6,4,3,7},
            {5,8,1,8},
            {7,6,9,4}
            };

        TaskAssignment assignment=new TaskAssignment(costMatrix);
        int minCost=assignment.findMinCost();
        System.out.println("最低成本为:"+minCost);
    }
}
```