
算法 1: 两道工序RGV工作流程(无故障)

Input: $t, updown[2], T, clean, location, Clocation, proportion, cncnow$

矩阵 t 表示RGV移动所需时间

$updown_1$ 表示RGV为奇数编号的CNC上下料所需时间

$updown_0$ 表示RGV为偶数编号的CNC上下料所需时间

矩阵 T 表示CNC加工完成一个两道工序物料的指定的工序所需时间

$clean$ 是RGV清洗时间

$location$ 表示RGV当前的位置

矩阵 $Clocation$ 表示CNC的位置

$proportion$ 表示加工第一道工序的CNC的个数(人为输入)

矩阵 $cncnow$ 表示当前CNC上面的工件编号

Output: ans

ans 表示最大加工数量

```
1 begin
    // 枚举工作第一道工序的CNC编号排列,找到最优解
2   arr = [1,2,3,4,5,6,7,8]
3   while next_permutation(arr, arr+size) do
4       solve(arr of porportion numsize)
5   end
6   return ans
7 end
8 void solve( int v[proportion])
9 {
10    ending , time , flag , RGVhand = 0
    // 初始上料
11    根据v的顺序依次给对应CNC上料
    // 根据智能调度算法进行模拟
12    while time ≤ 28800 do
13        调用RGV智能调度算法(2)得出下一个要处理的CNC编号k
        time = time + max( $t_{Clocation_k, location}, T_k$ )
14        location = Clocationk
15        RGVhand = cncnowk
16        cncnowk = flag?RGVhand : knew
17        time = time + updownk&1
18        if flag is true then
19            | time = time + clean ; ending = ending + 1
20        end
21        flag = flag ⊕ 1
22    end
23    ans = max(ans, ending)
24 }
```
