20174179 杨小川 计科卓越班

（1）n个零件需要加工，第i个零件加工时间为t[i]。现有m台完全一样的加工车床， 每个零件可以在任何一台机器加工且加工一旦开始就不会中断直至加工完成。求所有零件加工完成的最短时间。

（注：如果m=1，则结果等于 t[1]+t[2]+...+t[n])

首先考虑：

若 i < m，则最短时间为 MAX(t[1],t[2],...,t[n])

其次再考虑

若 m < i，则：根据贪心的原则

1 将t[1],t[2],...,t[n]按照时间长度降序排列。每次取出第一个（即时间最长的）零件放入剩下的车床中加工时间最少的。则最后一个时间最短的零件能放入最后加工时间最短的车床。

算法如下：

Let T[1..n] be a new array // 代表零件加工时间

Let A[1..m] be a new array // 代表加工车床

Sort(T) // 按照时间长短降序排列

for i = 1 -> n

temp = 1

for j = 1 -> m

if A[j] < A[temp]

temp = j

A[temp] += T[i]

#由贪心得出的为近似最优解。在某些情况下取不到最优解，例如：两台机器，零件耗时分别为10，8，3，3，3，3，3，3。此时最优解应为18，但贪心得出最优解为19。因此要求出绝对最优解应用暴力求解，时间复杂度为m^n，不可取，所以用贪心得到近似最优解较为合理。

1. 有n项任务，第i个任务的完成时间为t[i]，任务必须按照一定顺序依次完成，如果按照1，2，。。，n的顺序完成，则第i的任务完成时间为c[i] = t[1]+t[2]+...+t[i]，所有任务完成时间为c[1]+c[2]+...+c[n]。求所有任务完成时间最短的任务序列。

由于每项任务完成的时间代价都需要加上前面完成的任务的时间因此求最短时间的思路如下：

先将i个任务按照时间长短升序排序，再按照此顺序依次完成，即可获得最短完成时间

算法如下：

Let T[1..n] be a new array // 代表任务时间

Sort(T) // 按照时间长短升序排序

for i = 2 -> n

T[i] += T[i-1]

SUM(T[])