# Homework4 Solution

洪方舟
2016013259
Email: hongfz16@163.com

2018 年 3 月 24 日

## Problem 16-2

### a.

#### 算法设计：

将所有任务按照执行时间从小到大排序，就按照这个顺序执行，得到的平均运行结束时间是最小的

#### 正确性证明：

首先，观察到该问题有最优子结构，如果我们选择第一个执行的任务的时候就选择最优解，在选择第二个以及之后的执行任务顺序时，按照同样的方式选择执行任务的顺序，那么将会得到最优解。下面说明如果每次选择任务的时候贪心选择耗时最短的那一个，将会达到最优。设 $a$ 是当前待选择任务中耗时最短的那一个，$b$ 为剩下任务中任意一个，设解法 $A$ 将 $a$ 排在第一个，而解法 $B$ 将 $a$ 和 $b$ 的执行顺序对调，对于 $A$ 中 $b$ 之后执行的任务运行完成时间将没有变化，但是对于 $a$ 和 $b$ 两个任务之间的所有任务 $S$，由于在 $B$ 中首先需要运行时间较长的 $b$，所以 $S$ 中所有任务的平均完成时间在解法 $B$ 中将会长于解法 $A$，因此在每次选择下一个要执行的任务时，最优的方案就是贪心的选择耗时最短的任务。因此该算法具有正确性。

#### 时间复杂度分析：

只需要对所有任务按照执行时间从小到大排序即可，因此时间复杂度为 $O(nlgn)$

### b.

#### 算法设计：

使用最短剩余时间调度算法，伪代码如下

```
function SRTF(S)
    let tc=1, pc=\infty ,idc=-1
    let result[] be an empty array
    while not S.empty()
        for i in S.size()
            if tc \geq S[i].r and pc > S[i].p
                if pc \neq \infty and pc \neq 0
                    for j in range(0,S.size()-1)
```

```
                    if S[i].p \leq pc and pc \leq S[i+1].p
                        S.insert(s(r=0,p=pc),i)
                pc=S[i].p
                idc=i
                break
        tc+=1
        pc-=1
        result.append(idc)
    return result
```

**正确性证明：**

不难发现，该算法实际的完成顺序为 $S.r + S.p$ 的升序，如果重新构造一组任务 $S_n$，每个任务的执行时间对应为 $(S[i].r + S[i].p)$，并且使用上一问的无抢占的执行规则，则该算法产生的平均执行时间等于 $S_n$ 采用上一问的最优解计算得到的平均执行时间，则利用上一问的结论，可知在本小题有抢占的执行规则下，最短剩余时间调度算法能够实现平均执行时间最优。

**时间复杂度分析：**

如果令 $T$ 为完成所有任务所需的时间总长度，则外层 $while$ 需要循环 $T$ 次，内层 $for$ 循环耗时 $O(n)$，因此总的时间复杂度为 $O(nT)$

# Problem 16-5

**a.**

**b.**

**c.**