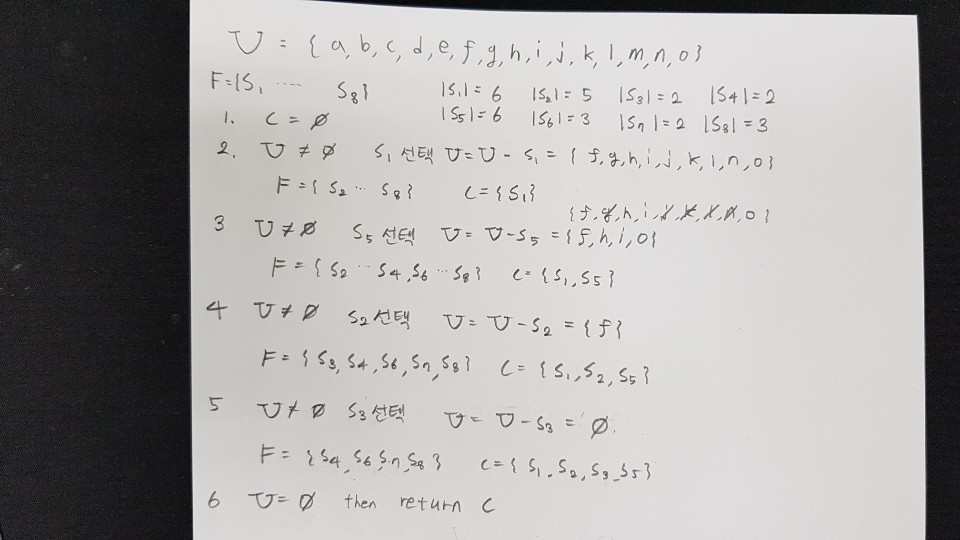
알고리즘 응용 #7과제

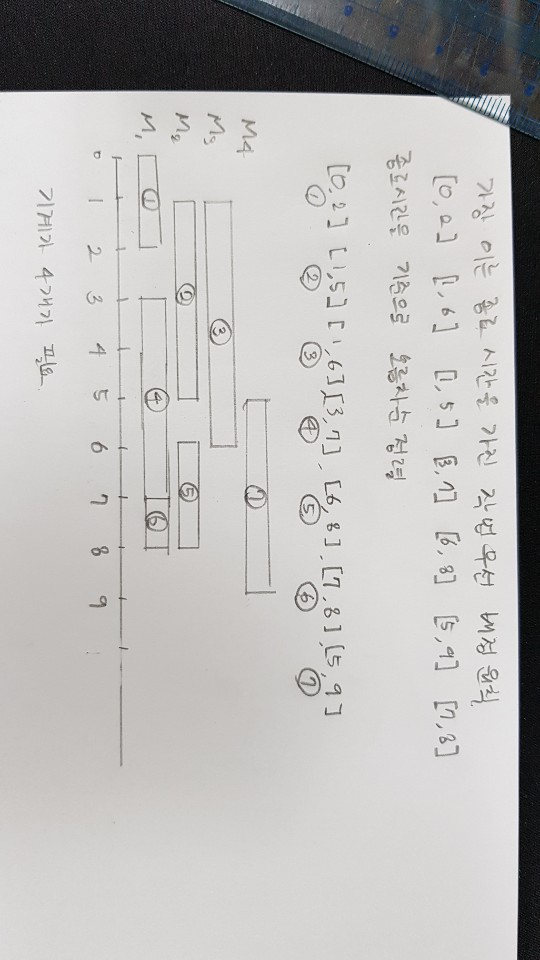
20144649 고한설

#1

S1 = {a,b,c,m,d,e} , S2 = {e,h,o,n,i} , S3 = {a,f} , S4 = {j,m}

S5 = {b,j,l,n,k,g} , S6 = {g,c,i}, S7 = {k,m} , S8 = {d,o,l}



#2

#3

**기계 1대에 가장 많은 수의 작업을 수행하려고 한다면**

**작업의 수행시간이 짧은 것들을 넣으면됩니다.**

종료시간-시작시간 = 각 작업의 걸리는 시간 ( k 라고 가정하겠습니다 )

K를 기준으로 작업을 오름차순으로 정렬합니다.

List <- 정렬한 리스트

While(List != 공집합) do

Ti <- List 에서 가장 짧은 작업 시간의 작업을 가져옵니다.

If(단 하나의 기계 M1이 Ti 를 수행할 수 있다면)

기계 M1에 Ti를 배정합니다.

Ti를 List에서 제거합니다.

Repeat

#4

문제 #3를 해결하기위해 이른 시작시간 작업 우선 배정원칙으로 작업을 배정하면 최적해를 항상 찾을 수 없습니다.

그 이유는 최악의 경우 가장 이른 시작시간을 가지는 작업이 가장 긴 작업일 경우를 생각해보면됩니다. 가장 이른시작지간을 가지므로 먼저 작업이 기계에 배정됩니다. 하지만 그 작업은 모든 작업 중 가장 긴 수행시간을 가지므로 기계에 단 하나의 작업만 배정되므로 최적해가 아닙니다. 따라서 항상 최적해를 찾을 수 없습니다.