FindMax(activity, upperbound)

max = 0

For i = 1 to activity.size-1

If max<activity[i].finishing and activity[i].finishing <= upperbound

max = activity[i].finishing

return max

greedy\_latest\_start

maxS = activity[0].starting

for i=1 to activity.size-1

if maxS<activity[i].starting

maxS = activity[i].starting

choose activity with maxS starting time // first activity chosen

while (true)

maxF = FindMax(activity, maxS) //find closest finishing time to the

//previous activity’s starting time

if maxF=0 // if maxF=0, means no more activity’s

break // finishing time is smaller

choose activity with maxF finishing time

maxS = starting time of the above activity