

Nome: Caio Silas de Araujo Amaro

Matrícula: 21.1.4111

- 1) Para o problema de seleção de atividades, prove que a atividade que começa por último faz parte de alguma solução ótima do problema.

Seja A_{ij} um subconjunto de tamanho máximo de atividades mutuamente compatíveis em S_{ij} e a_u é a atividade que começa por último de S_{ij} . Ordene as atividades em A_{ij} em ordem crescente de tempo de começo. Seja a_k a última atividade em A_{ij} .

Se $a_k = a_u$, então a_u já faz parte da solução ótima.

Caso contrário, construa $A'_{ij} = A_{ij} - \{a_k\} \cup \{a_u\}$.

- 2) Faça um algoritmo guloso para o problema de seleção de atividades, sendo que a escolha gulosa consiste em escolher primeiro a atividade que começa por último, ou seja, com o maior tempo de início.

Seja S um conjunto de atividades e A o conjunto solução.

Selecione a atividade que começa por último de S , chamada de a_u , e adicione-a ao conjunto A e remova-a de S . Remova do conjunto S todas as atividades que não são compatíveis com a atividade a_u , ou seja, todas as atividades que terminam depois de a_u ter começado. Repita esse processo enquanto S não estiver vazio.

```
function selecaoDeAtividades(S)
Entrada: o conjunto de atividades S
Saída: o subconjunto máximo de S de atividades mutuamente compatíveis

A = {}
while (S ≠ {}):
    a_u <- atividade que começa por último de S
    A <- A ∪ {a_u}
    S <- S - {a_u}
    S <- S - {atividades não compatíveis com a_u}

return A;
```