

**Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação e Estatística
Universidade de São Paulo**

Av. Trab. São Carlense, 400 - Centro, São Carlos - SP, 13566-590 – Fone (16) 3373-9700

RELATÓRIO PROJETOS

Algoritmos Avançados

Professor: João do E.S. Batista Neto

Alunos:

Matheus Tomieiro de Oliveira, 10734630

Victor Vieira Custodio Reis, 10734686

OUTUBRO/2019

Tasks with Deadline

O problema de tasks with deadline tem como meta resolver da melhor forma possível - sempre prezando por maximizar o lucro total -, o problema em que, temos vários prazos de entrega, cada um com um determinado lucro.

Com o programa em mãos, inicializa colocando quantas tarefas há na lista, e, para cada tarefa, o seu prazo de entrega e lucro, como mostrado a seguir:

5

3 40

3 10

3 20

5 10

5 70

Temos como saidas, uma lista das tarefas escolhidas e o lucro total adquirido, escolhendo as melhores tarefas:

Tarefas escolhidas:

task 1 deadline: 3 profit: 40

task 2 deadline: 3 profit: 10

task 3 deadline: 3 profit: 20

task 4 deadline: 5 profit: 10

task 5 deadline: 5 profit: 70

O lucro total foi 150

Para resolver o problema, foi utilizado uma solução gulosa que verifica, caso tenha mais de uma tarefa com o mesmo prazo de entrega, qual daquelas tarefas tem que se priorizada, levando a uma solução ótima local que leva a uma solução ótima global, isto é, se tiver outras tarefas que tenham o mesmo prazo de entrega. Isto é, o programa pega a lista de tarefas, ordena pelo prazo e passando pela lista, verifica se o prazo da tarefa é equivalente a data que ela deve ser entregue, se for, verifica se tem mais tarefas com o mesmo prazo e insere a que tiver o maior lucro; caso não for a data de entrega da tarefa, ela só insere na lista das tarefas já feitas. No final, imprime a lista de tarefas feitas e o lucro obtido.

Superstring Minima

O problema da superstring minima tem como meta, encontrar - dada uma lista de strings – a menor superstring formada por todos os elementos dessa lista.

Com o programa, passamos através de argumentos na chamada, várias string que formaram a lista de strings a serem trabalhadas.

Por exemplo, se usarmos a sintaxe:

```
lua Superstring.lua CATG CTAAGT GCTA TTCA ATGCATC
```

Receberemos então, como saída a seguinte Superstring: GCTAAGTTCATG

O paradigma utilizado na resolução desse tipo de problema, foi de solução gulosa. Tomamos em cada ponto, a decisão que melhor convém ao momento, chegando a uma solução global ótima. A cada novo ciclo, comparamos todos os elementos da lista entre si em suas permutações totais; assim, dentro de cada iteração avaliamos as strings, de duas em duas, comparando o inicio da primeira com o final da segunda(até cobrir todas as letras) e vice-versa. Ao final de cada ciclo, guardamos a string gerada(concatenação) na primeira posição da lista e operamos novamente. Ao final de todos os ciclos, sobrando somente a primeira posição da lista, retornamos ela própria.