

Trabalho 02 - Um escalonamento de três máquinas

Data de entrega: 08/07/2021

Importante:

- **Não** olhe códigos de outros grupos ou da internet. Exceto o que é fornecido.
- TODOS os membros do grupo devem participar e compreender completamente a implementação.
- Em caso de plágio, fraude ou tentativa de burlar o sistema será aplicado nota 0 na disciplina aos envolvidos.
- Alguns alunos podem ser solicitados para explicar com detalhes a implementação.
- Passar em todos os testes do run.codes não é garantia de tirar a nota máxima. Sua nota ainda depende do cumprimento das especificações do trabalho, qualidade do código, clareza dos comentários, boas práticas de programação e entendimento da matéria demonstrada em possível reunião.
- Você deverá submeter, até a data de entrega, o seu código na plataforma run.codes, onde o “Número de Matrícula” deverá ser o número do seu grupo. Somente um do grupo precisa submeter.

Esse trabalho deverá ser realizado em grupo, com os grupos já definidos na disciplina. Este trabalho deverá ser implementado em linguagem C.

Definição do Problema: Suponha que você tem 3 computadores idênticos, e uma coleção S de tarefas que você precisa processar, cada tarefa $s \in S$, tem um tempo de processamento t_s . Você deseja executar as tarefas no menor tempo possível, ou seja, o tempo que o computador mais carregado termina de processar a última tarefa. Formalmente seja M_i o tempo de término da máquina $i \in \{1, 2, 3\}$, você precisa colocar cada tarefa em uma das máquinas de forma minimizar o $\max\{M_1, M_2, M_3\}$.

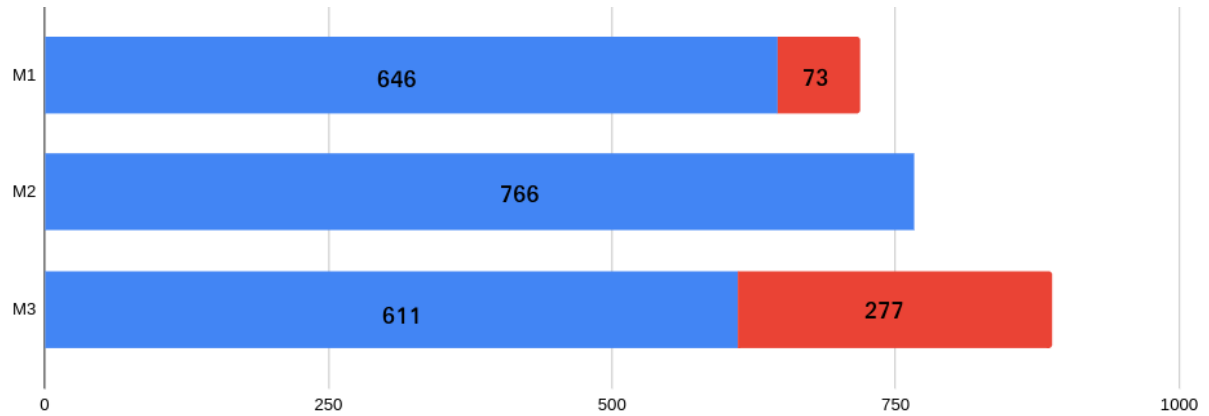
Seu programa deverá ler, da entrada padrão do sistema, um inteiro que diz quantas tarefas têm em S . Depois você tem $|S|$ linhas, cada uma com o tempo de processamento de uma atividade. Veja um exemplo de instância com 5 tarefas:

```
5
676
766
611
73
277
```

A saída esperada para seu programa será somente o tempo de término do computador com a maior carga:

```
888
```

Esse valor corresponde ao tempo de término da máquina três na seguinte solução:



- Seu algoritmo além de devolver a solução ótima, também deve ser rápido o bastante para passar nos testes do run.codes.
- Você não deve usar nenhuma biblioteca muito especializada.
- Se você não tiver certeza se alguma coisa é permitida ou não no trabalho, não hesite em perguntar ao professor!