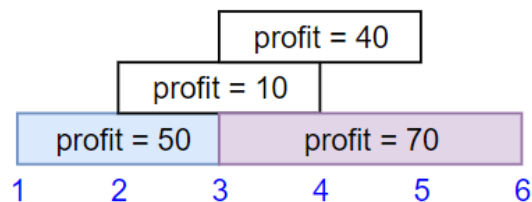


Trabalho 1 - Implementação de um Algoritmo de Escalonamento

Considere que você trabalha em uma Big Tech, que recebe jobs de diferentes empresas para serem executados o mais rápido possível. Sua tarefa será criar um algoritmo de escalonamento para esse conjunto de tarefas, observando as seguintes regras:

1. Existem N tarefas (jobs), onde cada tarefa deve ser escalonada considerando seus tempos de início e fim. A tarefa devolve ao sistema um determinado "lucro / benefício" se executada no tempo correto (ou seja, no tempo informado).
2. São informados os arrays `startTime`, `endTime` e `profit`. Este último retorna o lucro / benefício máximo que pode ser obtido de forma que não haja duas tarefas no subconjunto com intervalo de tempo sobreposto.
3. Ao escolher uma tarefa que termina no tempo X, uma outra tarefa que inicia no tempo X poderá ser selecionada. Ou seja, o tempo gasto para troca de contexto entre tarefas é desconsiderado.

Exemplo 1:

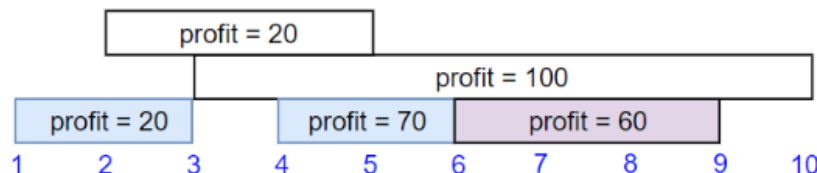


Entrada: `startTime = [1,2,3,3]`, `endTime = [3,4,5,6]`, `profit = [50,10,40,70]`

Saída: 120

Explicação: O subconjunto escolhido se refere ao primeiro e ao quarto job porque o intervalo de tempo $[1-3] + [3-6]$, retorna um lucro de 120 ($50 + 70$).

Exemplo 2:

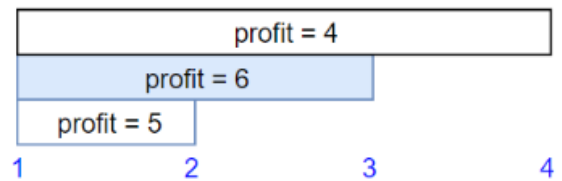


Entrada: `startTime = [1,2,3,4,6]`, `endTime = [3,5,10,6,9]`, `profit = [20,20,100,70,60]`

Saída: 150

Explicação: O subconjunto escolhido é o primeiro, quarto e quinto jobs. Lucro obtido 150 ($20 + 70 + 60$).

Exemplo 3:



Entrada: $startTime = [1, 1, 1]$, $endTime = [2, 3, 4]$, $profit = [5, 6, 4]$
Saída: 6

Os exemplos acima servem para auxiliar o entendimento do enunciado mas, para a elaboração do algoritmo, devem ser considerados todos os conjuntos de tarefas possíveis. Em outras palavras, o seu algoritmo deve atender qualquer conjunto de tarefas que cumpra as seguintes restrições:

```
1 <= startTime.length == endTime.length == profit.length <= 5 * 104
1 <= startTime[i] < endTime[i] <= 109
1 <= profit[i] <= 104
```

Você deverá implementar a sua solução usando a seguinte assinatura de método:

```
int jobScheduling(vector<int>& startTime, vector<int>& endTime, vector<int>& profit) {
    ...
}
```

O retorno do método é um valor inteiro que representa o lucro alcançado pela escala proposta pelo seu grupo.

ATENÇÃO: Esta assinatura não deve ser alterada em nenhuma hipótese. O trabalho não será corrigido em caso de alteração dessa assinatura.

Entrega: Um arquivo compactado contendo dois arquivos:

1. Um pdf com o relatório que mostra os principais pontos relacionados com a construção da solução;
2. Arquivo com código a ser compilado e executado.

Critérios de Avaliação:

- [4 pts] Clareza e organização do código desenvolvido.
 - [] Participação / entrega insuficiente para avaliar.
 - [] Ruim. Apresentou muita dificuldade de traduzir as ideias propostas para o código. Demonstra falta de familiaridade com a linguagem. A estrutura geral do código (modularização, variáveis) dificulta muito a leitura do código.
 - [] Regular. Demonstra alguma familiaridade com a linguagem, porém apresenta dificuldade em traduzir ideias para o código. A estrutura geral do código (modularização, variáveis) dificultam a leitura código.
 - [] Bom. Traduz ideias para código sem grandes dificuldades. A estrutura geral do código torna a leitura do mesmo fluida e fácil.
 - [] Ótimo. Demonstra grande fluência em traduzir ideias para código. Conhece recursos específicos ou avançados da linguagem. A estrutura geral do código torna a leitura fluída e fácil.
- [2 pts] Detalhamento da descrição dos pontos solicitados no enunciado do trabalho.
 - [] Participação / entrega insuficiente para avaliar.
 - [] Ruim. Mostrou conhecimento insuficiente dos fundamentos / conceitos apresentados.
 - [] Regular. Mostrou algum conhecimento acerca dos fundamentos/conceitos abordados, mas teve dificuldade em aplicá-los para resolver o problema proposto.
 - [] Bom. Mostrou conhecimento acerca dos fundamentos/conceitos abordados, conseguindo utilizá-los de forma adequada para resolver o problema proposta, mas com alguma dificuldade.
 - [] Ótimo. Mostrou domínio dos fundamentos/conceitos abordados, conseguindo aplicá-los na resolução de problemas sem dificuldades.

- [4 pts] Qualidade da explicação sobre o desenvolvimento da solução:
 - [] Participação / entrega insuficiente para avaliar.
 - [] Ruim. A solução proposta não produz resultados corretos;
 - [] Regular. Apresentou uma solução correta, ainda que não otimizada, mas não conseguiu discutir possíveis melhorias;
 - [] Bom. Conseguiu resolver o problema de forma não otimizada, mas conseguiu discutir com o(a) professor(a) como melhorar a solução proposta;
 - [] Ótimo. Conseguiu resolver o problema de forma otimizada, utilizando estruturas de dados e/ou algoritmos apropriados para o problema apresentado.

Observação: Trabalhos iguais receberão nota igual a zero.