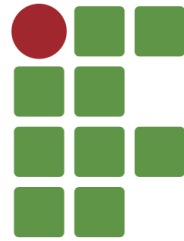


Estrutura de Dados e Algoritmos
Projeto 04: maratonando cursos
Ciência da Computação

Prof. Daniel Saad Nogueira Nunes



**INSTITUTO
FEDERAL**
Brasília

1 Contextualização

No período de isolamento social, Patrícia resolveu se capacitar e buscou cursos gratuitos de Computação online. Como esses cursos são curtos, ela consegue concluir qualquer um deles em exatamente uma semana e, para garantir a qualidade do aprendizado e não se sobrecarregar, Patrícia não faz dois ou mais cursos simultaneamente.

Cada curso tem a sua gratuidade limitada a algumas semanas e propicia à Patrícia uma certa quantidade de conhecimento. Como são muitos cursos e o número de semanas disponíveis que Patrícia tem até o período de isolamento chegar a seu fim é finito, ajude Patrícia indicando a ela os cursos que maximizam a quantidade total de conhecimento a ser adquirida neste período de isolamento social.

2 Especificação

O projeto deverá ser executado através da linguagem C.

A entrada deve ser lida da entrada padrão (`stdin`), enquanto a saída deverá ser impressa na saída padrão (`stdout`).

O programa deverá obedecer rigorosamente o formato de saída especificada, pois parte da correção será automatizada.

Obrigatoriamente a estrutura de dados **fila de prioridades** deverá ser utilizada.

2.1 Entrada

A primeira linha da entrada possui dois números inteiros separados por um espaço: N ($1 \leq N \leq 10^4$), o número de cursos gratuitos disponíveis e M ($1 \leq M \leq 10^3$), o número de semanas que Patrícia tem até o término do isolamento social.

As próximas N linhas descrevem os cursos. Cada linha possui dois inteiros separados por um espaço, G ($1 \leq G \leq M$) e V ($1 \leq V \leq 10^3$), que indicam, respectivamente, a quantidade de semanas de gratuidade de um curso e a quantidade de conhecimento proporcionada à Patrícia por este curso.

2.2 Saída

Imprima uma linha com um inteiro indicando a maior soma possível dos valores de conhecimento que Patrícia poderá adquirir no período de isolamento social.

3 Exemplos

- Entrada :

```
3 3
1 3
2 2
3 1
```

- Saída:

6

- Entrada:

3 3

1 5

1 3

3 2

- Saída:

7

- Entrada:

5 5

1 5

5 3

3 4

2 6

3 2

- Saída:

18

3.1 Compilação

Um arquivo `Makefile` deve ser disponibilizado para compilação do projeto.

3.2 Limites de Tempo e Memória

Para cada caso de teste, será permitido a execução do programa por apenas 1 segundo com utilização máxima de 256 MB de memória. Caso o programa leve mais tempo ou memória do que isso, será considerado que o algoritmo empregado foi ineficiente.

3.3 Documentação

Junto do(s) código(s) necessário(s) para resolver o problema, deverá ser disponibilizado um arquivo `README`, identificando o autor do trabalho e especificando as instruções para compilação e execução do(s) código(s).

3.4 Critérios de Correção

Fazem partes dos critérios de correção:

- Eficiência do programa.
- Utilização de estruturas de dados adequadas.
- Documentação: além do arquivo README, o código deve estar bem documentado.
- Legibilidade.

3.5 Ambiente de Correção

Os projetos serão corrigidos em uma máquina com sistema GNU/Linux e compilador `gcc 10.2.0`.

Trabalhos que não compilarem não serão avaliados.

4 Considerações

- Este projeto deve ser executado individualmente.
- Os trabalhos que incidirem plágio serão avaliados automaticamente com nota 0 para os envolvidos. Medidas disciplinares também serão tomadas.
- O trabalho deve ser entregue dentro de uma pasta zipada com a devida identificação do(s) aluno(s) através da sala de aula virtual da disciplina na data estipulada no ambiente.