

Sub - Vigilância em Cidades Inteligentes

Data de entrega: 13/08/2021 - meio-dia

Apresentações: 13/08/2021 - horário da aula

Importante:

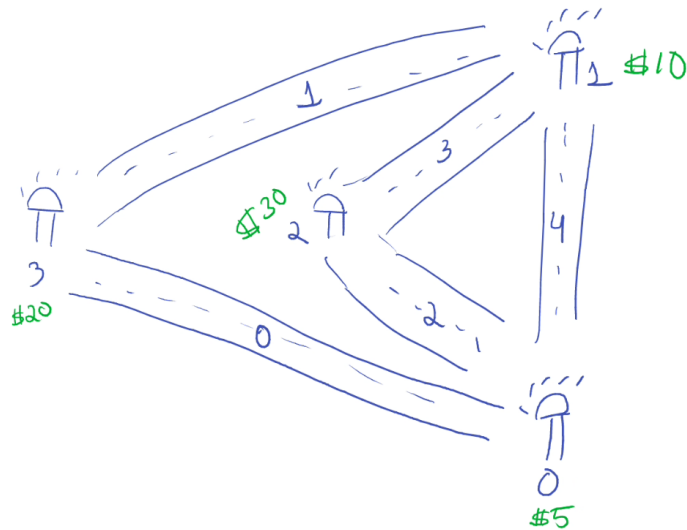
- **Não** olhe códigos de outros ou da internet.
- O trabalho deve ser feito completamente **INDIVIDUAL**.
- Em caso de plágio, fraude ou tentativa de burlar o sistema será aplicado nota 0 na disciplina.
- Passar em todos os testes do run.codes não é garantia de tirar a nota máxima. Sua nota ainda depende do cumprimento das especificações do trabalho, qualidade do código, clareza dos comentários, boas práticas de programação e entendimento da matéria demonstrada na apresentação.
- Você deverá submeter, até a data e horário de entrega, o seu código na plataforma run.codes.
- O programa deve **obrigatoriamente** ser feito em linguagem C.
- O programa deve **obrigatoriamente** passar em todos os casos do run.codes no tempo limite estipulado (os casos no run.codes serão disponibilizados até terça-feira)

Etapas da apresentação (20 minutos de apresentação + 10 de perguntas):

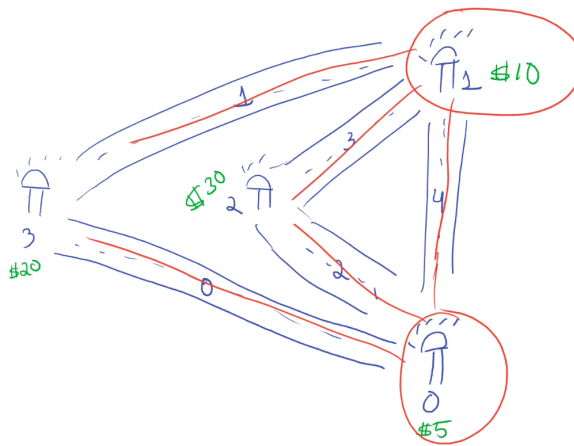
1. Demonstrar formalmente a qual classe pertence o problema (P, NP, NP-difícil, NP-completo)
2. Explicar com o máximo de detalhes o seu algoritmo, utilizando ilustrações.
3. Explicar sua implementação.
4. Arguição, etapa de perguntas.

Definição do Problema: Numa pequena cidade super moderna, os moradores decidiram instalar postes de vigilância em algumas esquinas, para vigiar **TODA** a cidade. Entretanto, a instalação de cada poste tem um custo diferente dependendo da esquina. E os moradores querem descobrir qual a solução mais barata. Formalmente temos um conjunto de ruas R e um conjunto P de postes possíveis. Cada poste $p \in P$ tem um custo $C(p)$ e cobre um subconjunto das ruas $S(p) \subseteq R$. O objetivo é encontrar um subconjunto $Q \subseteq P$, tal que

$$\bigcup_{p \in Q} S(p) = R, \text{ e } \sum_{p \in Q} C(p) \text{ seja mínimo.}$$



No exemplo acima temos 5 ruas e 4 postes, o poste 0 enxerga as ruas 0, 2 e 4; o poste 1 as ruas 1, 3 e 4; o poste 2 as ruas 2 e 3; e por fim o poste 3 enxerga as ruas 0 e 1. Os postes de 0 a 3 têm os seguintes custos: \$5, \$10, \$30 e \$20, nessa ordem. Uma solução para esse exemplo seria a instalação dos postes 0 e 1, que cobririam todas as ruas com custo de 15.



Seu programa deverá ler, da entrada padrão do sistema, dois inteiros que dizem quantos postes e quantas ruas a instância tem. Depois deverá ler $|P|$ linhas, cada uma com o índice do poste, o valor de instalação do poste, e a quantidade de ruas que ele cobre, seguido pelos índices das ruas (que vão de 0 até $|R| - 1$). O exemplo acima teria a seguinte entrada:

```
4 5
0 5 3 0 2 4
1 10 3 1 3 4
2 30 2 2 3
3 20 2 0 1
```

A saída esperada para seu programa será somente o custo da melhor solução (seguido de uma quebra de linha):

```
15
```