Análise e Desenho de Algoritmos

2º Teste 2021/22

18 de junho 2022

ERRATA

Caderno 1
Pergunta II.(a) Assuma que quando há mais que um caminho com comprimento mínimo entre s e t, esses caminhos não têm vértices em comum excepto os vértices s e t.

Por exemplo, se os dois caminhos seguintes são caminhos de comprimento mínimo com o valor d
$s, v_{i1}, v_{i2}, \ldots, v_{in}, t$
$s, v_{j1}, v_{j2}, \ldots, v_{jm}, t$

então para qualquer par de vértices (u,w) onde $u, w \in \{ v_{i1}, v_{i2}, \ldots, v_{in}, v_{j1}, v_{j2}, \ldots, v_{jm} \}$, temos $u \neq w$.

Número aluno: ________ Nome aluno: ____

Análise e Desenho de Algoritmos

2º Teste 2021/22

18 de junho 2022

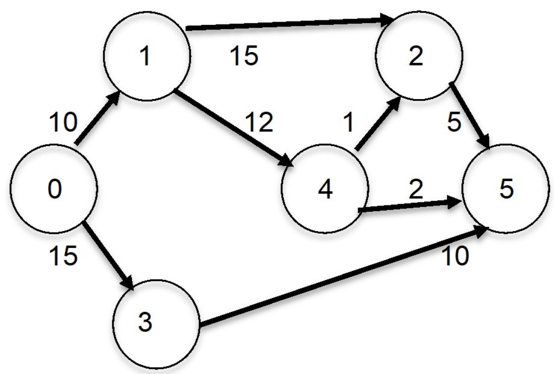
Caderno 1

Coloque o seu número e nome de forma legível em todas as folhas do teste.

Perguntas de implementação: se quiser usar algum método dos slides, não o copie; chame-o. Se quiser alterar algum método dos slides, escreva a sua nova definição integralmente.

Pergunta I (2 valores)

Considere o seguinte grafo Suponha que se executa o algoritmo de Dijkstra com este grafo e vértice de origem 0.



(a) Indique a ordem pela qual os vértices são seleccionados (retirados da fila com prioridade e passados a *exploreNode* como argumento).

(b) Identifique os arcos cujo tratamento termina numa execução do método *decreaseKey*.

(c) Indique o resultado da função *dijkstra*, ou seja, o conteúdo dos dois vectores retornados.

`length`:

0	1	2	3	4	5

`via`:

0	1	2	3	4	5

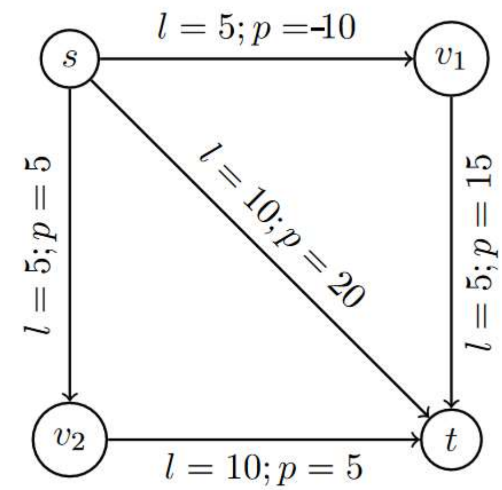
Número aluno: ________ Nome aluno: ____

Pergunta II (6 valores)

A Raquel, que é guia turística, está a organizar um passeio no campo num terreno acidentado (com subidas e descidas) para um grupo de pessoas em que algumas têm problemas de reumatismo.

O mapa do terreno pode ser representado com um grafo orientado *G*. O objectivo é ir do vértice origem, *s,* para o vértice destino, *t*. Cada arco (*u, v*) tem comprimento *l*(*u*, *v*) > 0 e declive *p*(*u,v*).

O seu objectivo será (1) encontrar um caminho com comprimento mínimo; e (2) entre todos os caminhos com comprimento mínimo, escolher o caminho com o declive total mínimo (somando todos os declives por onde se passa, queremos o caminho com o declive total mínimo).



Por exemplo, no terreno representado com o grafo da figura, há três caminhos de *s* para *t* . Iríamos escolher o caminho *s,* v_1*, t* porque tem comprimento mínimo (10) e de entre os caminhos com comprimento mínimo, é o que tem o declive total mínimo (5). O caminho *s, t* também tem comprimento 10 mas não é escolhido porque o seu declive total é 20.

Assuma que o comprimento *l* é devolvido pelo método *label*. Assuma também que estes arcos têm um atributo extra, o declive *p*, e que esse declive é devolvido pelo método *label2.* Por exemplo, se *e* representa o arco (*s,t*) , *e.label*() devolve 10 e *e.label2*() devolve 20.

(a) Escreva um algoritmo (em pseudo-código) que devolve o caminho pretendido (mais curto e com o menor declive total). O algoritmo recebe o grafo, o vértice origem e o vértice destino.

NOTA: Lembre-se que se quiser usar algum método dos slides, não o copie; chame-o. Se quiser alterar algum método dos slides, escreva a sua nova definição integralmente.

(b) Estude (justificando) a complexidade temporal do seu algoritmo, no pior caso.

Use o espaço em baixo para responder às duas alíneas.

Número aluno: ________ Nome aluno: ____

Análise e Desenho de Algoritmos

2º Teste 2021/22

18 de junho 2022

Caderno 2

Coloque o seu número e nome de forma legível em todas as folhas do teste.

Perguntas de implementação: se quiser usar algum método dos slides, não o copie; chame-o. Se quiser alterar algum método dos slides, escreva a sua nova definição integralmente.

Pergunta III (6 valores)

O Professor Chopin, que é professor de piano, decidiu fazer uma audição com os seus alunos. O Prof. Chopin tem *n* alunos e cada um sabe tocar bem três músicas (de um total de *m* músicas). Nesta audição, cada aluno irá tocar apenas uma música (das três que sabe tocar bem).

O Prof. Chopin quer saber se é possível fazer a audição sem repetição de músicas. Ou seja, se todos os alunos participarem, consegue-se fazer uma audição em que não haverá músicas repetidas?

Exemplo 1:

O João sabe tocar bem as músicas m1, m2 e m3,
a Joana sabe tocar bem as músicas m1, m3 e m4,
a Maria sabe tocar bem as músicas m1, m2 e m4, e
o Carlos sabe tocar bem as músicas m1, m2 e m3.
Neste caso consegue-se fazer uma audição sem repetição de músicas porque o João pode tocar m1, a Joana m3, a Maria m4 e o Carlos m2.

Exemplo 2:

O João, a Joana, a Maria e o Carlos sabem tocar bem as músicas m1, m2 e m3.
Neste caso não se consegue fazer uma audição sem repetição de músicas.

(a) **Desenhe** o grafo(s) que construiria para resolver este problema com o **<u>exemplo 1</u>**.

(b) Escreva um algoritmo (em pseudo-código) que retorna *true* se, e só se, é possível fazer uma audição em que todos os alunos participam sem que haja repetição de músicas.

O algoritmo recebe o número de músicas, *m*, e uma matriz, *mat*, de *n* por 3. Cada uma das *n* linhas corresponde a um dos *n* alunos e tem informação sobre quais as músicas que esse aluno sabe tocar bem. Por exemplo, a matriz do exemplo 1 seria a seguinte:

m1	m2	m3
m1	m3	m4
m1	m2	m4
m1	m2	m3

Na sua implementação, pode usar o método *buildGraph* (não precisa de implementar este método) que recebe *m* e a matriz *mat*, e devolve um grafo semelhante ao que ilustrou na alínea anterior (mas adaptado aos dados presentes na matriz).

Número aluno: _________ Nome aluno: ____

NOTA: Lembre-se que se quiser usar algum método dos slides, não o copie; chame-o. Se quiser alterar algum método dos slides, escreva a sua nova definição integralmente.

Use o espaço em baixo para responder.

Pergunta IV (1 valor)
Suponha que há uma redução polinomial do problema A para o problema B ($A \leq_P B$). Quais das seguintes afirmações ***têm*** que ser verdadeiras?
(Marque *sim* nas afirmações que **têm** que ser verdadeiras, caso contrário marque *não*. Respostas correctas valem 0.25 valores. Respostas erradas descontam 0.15 valores.)

O problema B é NP-difícil.	Sim	Não
Um algoritmo de tempo polinomial para B pode ser usado para resolver A em tempo polinomial.	Sim	Não
Se não há nenhum algoritmo de tempo polinomial para B, então também não existe nenhum algoritmo de tempo polinomial para A.	Sim	Não
Se A é NP-difícil e há um algoritmo de tempo polinomial para B, então P = NP.	Sim	Não
Se B é NP-difícil então A é NP-difícil.	Sim	Não
Se B é decidível então A é decidível.	Sim	Não

Número aluno: ________ Nome aluno: ____

Pergunta V (5 valores)

Num grafo, um ciclo simples é um ciclo sem vértices repetidos (excepto o primeiro e último vértice, que são o mesmo vértice).

Considere o seguinte problema, a que vamos chamar **problema do ciclo n/2**:

Dado um grafo *G*, *G* tem um ciclo simples de comprimento igual a *floor*(*n*/2)? (Lembre-se que a função *floor* arredonda "para baixo".)

(a) Prove que o problema do ciclo n/2 é NP.

(b) Prove que o problema do ciclo n/2 é NP-difícil.

(c) O problema do ciclo n/2 é NP-completo? Justifique a sua resposta, assumindo que resolveu as duas alíneas anteriores.