Universidade do Porto — L:EIC

Desenho de Algoritmos — 2021/22

Exame (16.07.2022) - Parte Prática — *90 minutos*

N.º [   ] Nome [   ]

Atenção: Não serão prestados esclarecimentos durante a prova. Leia pf as perguntas com atenção. Deve responder em **quatro folhas**:

- 1ª Folha: **1a), 1b)**.
- 2ª Folha: **1c), 1d)**
- 3ª Folha: **1 e), 1f)**.
- 4ª Folha: **2**.

1. Uma associação fretou m veículos idênticos a uma empresa para realizar uma excursão e identificou-os por números de 0 a $m-1$. Todos têm capacidade c, sem incluir o/a motorista. Algumas pessoas gostariam de viajar em grupo, no mesmo veículo, e manifestaram essa vontade na reserva. Tal informação foi processada, resultando em n grupos, alguns com um só elemento. Os grupos foram numerados de 0 a $n-1$. O número de elementos do grupo k é d_k, sendo $d_k \geq 1$, para $0 \leq k < n$. A associação pretende atribuir os grupos aos veículos de modo que **o número máximo de pessoas por veículo seja mínimo**. Nenhum grupo se separa.

Um membro da associação, delineou um algoritmo simples: *"ordenar os grupos por ordem decrescente de d_k (se houver empates, escolher um qualquer); a seguir, efetuar a atribuição segundo essa ordem, usando sempre o veículo que estiver menos ocupado (se houver empates, escolher um qualquer)."*

a) [1.5 v] Assumindo que os grupos tinham sido numerados de modo que $d_0 \geq d_1 \geq \ldots \geq d_{n-1}$, implemente **em C/C++ ou Java** tal estratégia como uma função que determina a **solução** (i.e., o veículo atribuído a cada grupo) e o **número de máximo** de lugares ocupados (nalgum veículo). Se não for possível atribuir lugares a todos os grupos, deve indicar -1 em vez desse número. A função terá n, m, c e d como argumentos, além de outros que possa definir para o resultado. Não deve imprimir nada mas pode retornar valores.

b) [1.0 v] Indique a **complexidade temporal** da sua implementação, justificando sucintamente.

c) [1.0 v] Apresente **duas instâncias** em que o algoritmo falha: **(1)** por alguns grupos ficarem sem lugar, embora fosse possível atribuir lugares a todos; **(2)** por produzir uma solução que não é ótima.

d) [1.5 v] Admitindo que a estratégia apresentada pelo membro da associação produzia uma solução, de que modo tal solução poderia ser útil na pesquisa de uma solução ótima por um método exato, como *pesquisa com retrocesso* (*backtracking*) e *branch-and-bound* (para programação inteira) ?

e) [1.0 v] Recorrendo ao problema de decisão PARTITION, que é NP-completo, prove que, se P $\neq$ NP, o problema enunciado não pode ser resolvido por um algoritmo polinomial, mesmo se $m = 2$.

f) [1.5 v] Formule **matematicamente** o problema. Como **variáveis de decisão** deve usar $x_{ij} \in \{0, 1\}$, que indicará se o grupo i viaja no veículo j, além de outras variáveis que possam ser úteis. Indique a interpretação das **restrições** e da **função objetivo**. As restrições devem ser **lineares** e garantir a atribuição de lugares a todos os grupos, se for possível.

(Continua, v.p.f.)

Universidade do Porto - L:EIC - Desenho de Algoritmos – Exame 16.07.2022

N.º | Nome

2. Para a excursão, todos os veículos irão partir de um mesmo local s e terão destino t. Todos seguirão o mesmo trajeto. Com receio de que o ar condicionado avarie, a associação analisou os valores das temperaturas previstas para todos os troços da rede viária, e pretende encontrar um **caminho (percurso acíclico) de s para t em que a temperatura máxima seja mínima**.

a) $\boxed{\text{1.5 v}}$ Usando **pseudocódigo** (C/C++ ou Java), implemente uma função para resolver o problema, que se baseie numa abordagem semelhante à do algoritmo de Dijkstra, mas **adaptada convenientemente**. A função deve obter um **caminho ótimo** ("ótimo", no sentido definido) e a **temperatura máxima** nesse caminho. Para representar o caminho, pode indicar qual é o nó que precede cada nó no caminho.

Admita que a rede viária é dada por um grafo dirigido $G = (V, E, T)$, em que $T(x, y) \in \mathbb{Z}^+$ indica a temperatura no ramo $(x, y) \in E$. Embora não seja relevante, pode assumir que $T(x, y) = T(y, x)$, se $(x, y) \in E$ e $(y, x) \in E$. Os identificadores de s e t são passados como argumentos, além de G.

b) $\boxed{\text{1.5 v}}$ Indique a **complexidade temporal** da função, justificando sucintamente. Que técnica(s) de concepção de algoritmos aplica?

c) $\boxed{\text{1.5 v}}$ **Ilustre o comportamento da função** na instância seguinte, descrevendo de forma concisa e sucinta o que acontece em cada passo (pode usar anotações no grafo e indicar a ordem pela qual vai explorando os nós). Pode responder a esta questão mesmo que não tenha respondido a **2a)**, mas o algoritmo que aplicar terá de ser baseado numa adaptação do algorimo de Dijkstra.